10/804,097

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。
This is to certify that the annexed is a true copy of the following application as filed with this Office.

願年月日 Date of Application:	２００３年　３月３１日
願番号 Application Number:	特願２００３－０９６１２９
[ST. 10/C]:	[ＪＰ２００３－０９６１２９]
願人 Applicant(s):	株式会社リコー

CERTIFIED COPY OF
PRIORITY DOCUMENT

BEST AVAILABLE COPY

２００４年　４月１２日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫

出証番号　出証特２００４－３０２９７５８

【書類名】　特許願
【整理番号】　0302634
【提出日】　平成15年 3月31日
【あて先】　特許庁長官　　殿
【国際特許分類】　H04L 9/14
【発明の名称】　デジタル証明書管理システム、デジタル証明書管理装置、デジタル証明書管理方法、更新手順決定方法およびプログラム
【請求項の数】　28
【発明者】
【住所又は居所】　東京都大田区中馬込１丁目３番６号　株式会社リコー内
【氏名】　榎田　寛朗
【特許出願人】
【識別番号】　000006747
【住所又は居所】　東京都大田区中馬込１丁目３番６号
【氏名又は名称】　株式会社リコー
【代表者】　桜井　正光
【代理人】
【識別番号】　100080931
【住所又は居所】　東京都豊島区東池袋１丁目２０番２号　池袋ホワイトハウスビル８１８号
【弁理士】
【氏名又は名称】　大澤　敬
【手数料の表示】
【予納台帳番号】　014498
【納付金額】　21,000円
【提出物件の目録】
【物件名】　明細書　1
【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9809113

【プルーフの要否】　要

【書類名】　　明細書

【発明の名称】　デジタル証明書管理システム、デジタル証明書管理装置、デジタル証明書管理方法、更新手順決定方法およびプログラム

【特許請求の範囲】

【請求項１】　１又は複数のクライアントと１又は複数のサーバとによって構成され、該各クライアントと各サーバとの間でデジタル証明書を用いて相互認証を行うようにしたクライアント・サーバシステムに、前記各クライアント及び前記各サーバと通信可能なデジタル証明書管理装置を接続したデジタル証明書管理システムであって、

前記デジタル証明書管理装置に、

前記各クライアント及び前記各サーバが前記相互認証に使用する前記デジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、

前記クライアント・サーバシステムを構成する各ノードについて、該ノードの通信相手及び該通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶する構成記憶手段と、

該構成記憶手段に記憶している情報をもとに、前記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段とを設け、

前記証明鍵更新手段に、

更新用の新証明鍵を取得する手段と、

該新証明鍵を用いて正当性を確認可能な、前記相互認証に使用するための新デジタル証明書を取得する手段と、

前記各クライアントのための新デジタル証明書である新クライアント証明書と、前記新証明鍵とをそれぞれ対応する前記各クライアントに送信してこれを記憶するよう要求する第１の更新要求手段と、

前記各サーバのための新デジタル証明書である新サーバ証明書と、前記新証明鍵とをそれぞれ対応する前記各サーバに送信してこれを記憶するよう要求する第２の更新要求手段とを設け、

前記更新順制御手段が、前記第２の更新要求手段がそれぞれの前記サーバに前記新サーバ証明書を送信してこれを記憶するよう要求する動作を、該サーバの通

信相手となる全てのクライアントから前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理システム。

【請求項２】　請求項１記載のデジタル証明書管理システムであって、

前記デジタル証明書管理装置の前記更新順制御手段が、前記第１の更新要求手段がそれぞれの前記クライアントに前記新クライアント証明書を送信してこれを記憶するよう要求する動作を、該クライアントの通信相手となる全てのサーバから前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理システム。

【請求項３】　請求項１記載のデジタル証明書管理システムであって、

前記デジタル証明書管理装置の前記更新順制御手段が、前記第１の更新要求手段が前記新クライアント証明書と前記新証明鍵とを同時に前記各クライアントに送信してこれらを記憶するよう要求し、前記第２の更新要求手段が、それぞれの前記サーバに対して、該サーバの通信相手となる全てのクライアントから前記新証明鍵を記憶した旨の応答があった後で、前記新サーバ証明書と前記新証明鍵とを同時に送信し、これらを記憶するよう要求するように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理システム。

【請求項４】　前記各サーバに、前記デジタル証明書管理装置と少なくとも一つの前記クライアントとの間の通信を仲介する手段を設け、

前記デジタル証明書管理装置と前記各クライアントとはいずれかの前記サーバを介して通信を行うことを特徴とする請求項１乃至３のいずれか一項記載のデジタル証明書管理システム。

【請求項５】　前記各クライアントに、前記デジタル証明書管理装置と少なくとも一つの前記サーバとの間の通信を仲介する手段を設け、

前記デジタル証明書管理装置と前記各サーバとはいずれかの前記クライアントを介して通信を行うことを特徴とする請求項１乃至３のいずれか一項記載のデジタル証明書管理システム。

【請求項６】　前記各クライアントに、前記デジタル証明書管理装置と該クライアントとの間の通信を仲介するサーバに対して定期的に通信を要求する手段

を設け、

前記サーバから前記クライアントへ送信すべき情報は、該通信の要求に対する応答として送信するようにしたことを特徴とする請求項４記載のデジタル証明書管理システム。

【請求項７】　請求項１乃至６のいずれか一項記載のデジタル証明書管理システムであって、

前記デジタル証明書管理装置の前記証明鍵更新手段に、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む証明鍵証明書を取得する手段を設け、

前記第１の更新要求手段は、前記新証明鍵を前記証明鍵証明書の形式で前記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求する手段であり、

前記第２の更新要求手段は、前記新証明鍵を前記証明鍵証明書の形式で前記各サーバに送信してここに含まれる証明鍵を記憶するよう要求する手段であり、

前記各クライアント及び前記各サーバにそれぞれ、

前記デジタル証明書管理装置から前記証明鍵証明書に含まれる証明鍵の記憶を要求された場合に、受信した証明鍵証明書の正当性を従前の証明鍵を用いて確認し、そこに含まれる証明鍵が適当なものであると判断した場合に該証明鍵を記憶する手段を設けたことを特徴とするデジタル証明書管理システム。

【請求項８】　請求項１乃至６のいずれか一項記載のデジタル証明書管理システムであって、

前記デジタル証明書管理装置の前記証明鍵更新手段に、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第１の証明鍵証明書を取得する手段と、

前記新証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第２の証明鍵証明書を取得する手段とを設け、

前記第１の更新要求手段は、前記新証明鍵を前記第１及び第２の証明鍵証明書の形式でそれぞれ前記各クライアントに送信してこれを記憶するよう要求する手段であり、

前記第２の更新要求手段は、前記新証明鍵を前記第１及び第２の証明鍵証明書の形式でそれぞれ前記各サーバに送信してこれを記憶するよう要求する手段であり、

前記各クライアント及び前記各サーバにそれぞれ、

前記デジタル証明書管理装置から前記第１の証明鍵証明書を記憶するよう要求された場合に、該証明書の正当性を従前の証明鍵を用いて確認し、これが適当なものであると判断した場合に該証明書を記憶する手段と、

前記デジタル証明書管理装置から前記第２の証明鍵証明書を記憶するよう要求された場合に、該証明書の正当性を前記第１の証明鍵証明書に含まれる前記新証明鍵を用いて確認し、前記第２の証明鍵証明書が適当なものであると判断した場合に、該証明書を記憶すると共に従前の証明鍵証明書及び前記第１の証明鍵証明書を削除する手段とを設け、

前記デジタル証明書管理装置の前記更新順制御手段が、前記第１の更新要求手段が前記第２の証明鍵証明書をそれぞれの前記クライアントに送信してこれを記憶するよう要求する動作を、少なくとも該クライアントの通信相手となる全てのサーバから前記新サーバ証明書を記憶した旨の応答があった後に行うよう制御し、前記第２の更新要求手段が前記第２の証明鍵証明書をそれぞれの前記サーバに送信してこれを記憶するよう要求する動作を、少なくとも該サーバの通信相手となる全てのクライアントから前記新クライアント証明書を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理システム。

【請求項９】　請求項１乃至８のいずれか一項記載のデジタル証明書管理システムであって、

前記クライアントと前記サーバが行う前記相互認証は、ＳＳＬ又はＴＬＳのプロトコルに従った相互認証であり、

前記クライアント証明書及び前記サーバ証明書はそれぞれ前記各クライアント及び前記各サーバの公開鍵証明書であることを特徴とするデジタル証明書管理システム。

【請求項１０】　クライアント・サーバシステムを構成する１又は複数のク

ライアント及び１又は複数のサーバと通信可能なデジタル証明書管理装置であって、

前記各クライアントと前記各サーバとの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、

前記クライアント・サーバシステムを構成する各ノードについて、該ノードの通信相手及び該通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶する構成記憶手段と、

該構成記憶手段に記憶している情報をもとに、前記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段とを設け、

前記証明鍵更新手段に、

更新用の新証明鍵を取得する手段と、

該新証明鍵を用いて正当性を確認可能な、前記相互認証に使用するための新デジタル証明書を取得する手段と、

前記クライアントのための新デジタル証明書である新クライアント証明書と、前記新証明鍵とをそれぞれ前記クライアントに送信してこれを記憶するよう要求する第１の更新要求手段と、

前記サーバのための新デジタル証明書である新サーバ証明書と、前記新証明鍵とをそれぞれ前記サーバに送信してこれを記憶するよう要求する第２の更新要求手段とを設け、

前記更新順制御手段が、前記第２の更新要求手段がそれぞれの前記サーバに対して前記新サーバ証明書を送信してこれを記憶するよう要求する動作を、該サーバの通信相手となる全てのクライアントからの前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理装置。

【請求項１１】　請求項１０記載のデジタル証明書管理装置であって、

前記更新順制御手段が、前記第１の更新要求手段がそれぞれの前記クライアントに前記新クライアント証明書を送信してこれを記憶するよう要求する動作を、該クライアントの通信相手となる全てのサーバからの前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴と

するデジタル証明書管理装置。

【請求項１２】　請求項１０記載のデジタル証明書管理装置であって、

前記更新順制御手段が、前記第１の更新要求手段が前記新クライアント証明書と前記新証明鍵とを同時に前記各クライアントに送信してこれらを記憶するよう要求し、前記第２の更新要求手段が、それぞれの前記サーバに対して、該サーバの通信相手となる全てのクライアントからの前記新証明鍵を記憶した旨の応答があった後で、前記新サーバ証明書と前記新証明鍵とを同時に送信し、これらを記憶するよう要求するように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理装置。

【請求項１３】　請求項１０乃至１２のいずれか一項記載のデジタル証明書管理装置であって、

前記証明鍵更新手段に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む証明鍵証明書を取得する手段を設け、

前記第１の更新要求手段は、前記新証明鍵を前記証明鍵証明書の形式で前記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求する手段であり、

前記第２の更新要求手段は、前記新証明鍵を前記証明鍵証明書の形式で前記各サーバに送信してここに含まれる証明鍵を記憶するよう要求する手段であることを特徴とするデジタル証明書管理装置。

【請求項１４】　請求項１０乃至１２のいずれか一項記載のデジタル証明書管理装置であって、

前記証明鍵更新手段に、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第１の証明鍵証明書を取得する手段と、

前記新証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第２の証明鍵証明書を取得する手段とを設け、

前記第１の更新要求手段が、前記新証明鍵を前記第１及び第２の証明鍵証明書の形式でそれぞれ前記各クライアントに送信してこれを記憶するよう要求する手段であって、前記各クライアントに、前記第２の証明鍵証明書を記憶する場合に

従前の証明鍵証明書及び前記第１の証明鍵証明書を削除させる手段を有し、

前記第２の更新要求手段が、前記新証明鍵を前記第１及び第２の証明鍵証明書の形式でそれぞれ前記サーバに送信してこれを記憶するよう要求する手段であって、前記サーバに、前記第２の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び前記第１の証明鍵証明書を削除させる手段を有し、

前記更新順制御手段が、前記第１の更新要求手段が前記第２の証明鍵証明書をそれぞれの前記クライアントに送信してこれを記憶するよう要求する動作を、少なくとも該クライアントの通信相手となる全てのサーバから前記新サーバ証明書を記憶した旨の応答があった後に行い、前記第２の更新要求手段が前記第２の証明鍵証明書をそれぞれの前記サーバに送信してこれを記憶するよう要求する動作を、少なくとも該サーバの通信相手となる全てのクライアントから前記新クライアント証明書を記憶した旨の応答があった後に行うように前記更新手順を制御する手段であることを特徴とするデジタル証明書管理装置。

【請求項１５】　請求項１０乃至１４のいずれか一項記載のデジタル証明書管理装置であって、

前記相互認証は、ＳＳＬ又はＴＬＳのプロトコルに従った相互認証であり、

前記クライアント証明書及び前記サーバ証明書はそれぞれ前記各クライアント及び前記各サーバの公開鍵証明書であることを特徴とするデジタル証明書管理装置。

【請求項１６】　クライアント・サーバシステムを構成する１又は複数のクライアントと１又は複数のサーバとの間で相互認証に使用するデジタル証明書を、前記各クライアント及び前記各サーバと通信可能なデジタル証明書管理装置によって管理するデジタル証明書管理方法であって、

前記デジタル証明書管理装置が、

前記クライアント・サーバシステムを構成する各ノードについて、該ノードの通信相手及び該通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶しておき、該情報をもとに定める更新手順に従って前記各クライアント及び前記各サーバが前記相互認証に使用する前記デジタル証明書の正当性を確認するための証明鍵を更新し、

該証明鍵の更新を、

更新用の新証明鍵を取得する手順と、

該新証明鍵を用いて正当性を確認可能な、前記相互認証に使用するための新デジタル証明書を取得する手順と、

前記各クライアントのための新デジタル証明書である新クライアント証明書と、前記新証明鍵とをそれぞれ対応する前記各クライアントに送信してこれを記憶させる手順と、

前記各サーバのための新デジタル証明書である新サーバ証明書と、前記新証明鍵とをそれぞれ対応する前記各サーバに送信してこれを記憶させる手順とを実行することによって行い、

前記更新手順を、それぞれの前記サーバに前記新サーバ証明書を送信してこれを記憶させる手順を該サーバの通信相手となる全てのクライアントから前記新証明鍵を記憶した旨の応答があった後に行うよう定めることを特徴とするデジタル証明書管理方法。

【請求項１７】　請求項１６記載のデジタル証明書管理方法であって、

前記更新手順を、それぞれの前記クライアントに前記新クライアント証明書を送信してこれを記憶させる手順を該クライアントの通信相手となる全てのサーバから前記新証明鍵を記憶した旨の応答があった後に行うよう定めることを特徴とするデジタル証明書管理方法。

【請求項１８】　請求項１６記載のデジタル証明書管理方法であって、

前記更新手順を、前記新クライアント証明書と前記新証明鍵とを同時に前記各クライアントに送信してこれらを記憶させるように定め、さらに、それぞれの前記サーバに対して、該サーバの通信相手となる全てのクライアントから前記新証明鍵を記憶した旨の応答があった後で、前記新サーバ証明書と前記新証明鍵とを同時に送信し、これらを記憶させるように定めることを特徴とするデジタル証明書管理方法。

【請求項１９】　請求項１６乃至１８のいずれか一項記載のデジタル証明書管理方法であって、

前記証明鍵の更新の際に、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む証明鍵証明書を取得する手順をさらに実行し、

前記新証明鍵を前記各サーバあるいは前記各クライアントに送信してこれを記憶させる手順において、該新証明鍵を前記証明鍵証明書の形式で送信してここに含まれる証明鍵を記憶させるようにし、

前記各クライアント又は前記各サーバに前記証明鍵証明書に含まれる証明鍵を記憶させる場合に、該証明鍵証明書の正当性を、記憶している従前の証明鍵を用いて確認させ、そこに含まれる証明鍵が適当なものであると判断した場合に該証明鍵を記憶させることを特徴とするデジタル証明書管理方法。

【請求項２０】　請求項１６乃至１８のいずれか一項記載のデジタル証明書管理方法であって、

前記証明鍵の更新の際に、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第１の証明鍵証明書を取得する手順と、

前記新証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第２の証明鍵証明書を取得する手順とをさらに実行し、

前記新証明鍵を前記各サーバあるいは前記各クライアントに送信してこれを記憶させる手順において、該新証明鍵を前記第１及び第２の証明鍵証明書の形式でそれぞれ送信してこれを記憶させるようにし、

前記更新手順を、前記第２の証明鍵証明書をそれぞれの前記クライアントに送信してこれを記憶するよう要求する手順を少なくとも該クライアントの通信相手となる全てのサーバから前記新サーバ証明書を記憶した旨の応答があった後に行うよう定め、さらに、前記第２の証明鍵証明書をそれぞれの前記サーバに送信してこれを記憶するよう要求する動作を少なくとも該サーバの通信相手となる全てのクライアントから前記新クライアント証明書を記憶した旨の応答があった後に行うよう定め、

前記各クライアント又は前記各サーバに前記第１の証明鍵証明書を記憶させる際に、該証明書の正当性を従前の証明鍵を用いて確認させ、これが適当なものであると判断した場合に該証明書を記憶させ、

前記各クライアント又は前記各サーバに前記第２の証明鍵証明書を記憶させる際に、該証明書の正当性を前記第１の証明鍵証明書に含まれる前記新証明鍵を用いて確認させ、前記第２の証明鍵証明書が適当なものであると判断した場合に、該証明書を記憶させると共に従前の証明鍵証明書及び前記第１の証明鍵証明書を削除させることを特徴とするデジタル証明書管理方法。

【請求項２１】　請求項１６乃至２０のいずれか一項記載のデジタル証明書管理方法であって、

前記クライアントと前記サーバとの間の前記相互認証は、ＳＳＬ又はＴＬＳのプロトコルに従った相互認証であり、

前記クライアント証明書及び前記サーバ証明書はそれぞれ前記各クライアント及び前記各サーバの公開鍵証明書であることを特徴とするデジタル証明書管理方法。

【請求項２２】　クライアント・サーバシステムを構成する１又は複数のクライアントと１又は複数のサーバとに記憶させ、これらの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を、前記各クライアント及び前記各サーバと通信可能なデジタル証明書管理装置によって更新する際の更新手順を定める更新手順決定方法であって、

前記デジタル証明書管理装置が、

前記クライアント・サーバシステムを構成する各ノードについて、該ノードの通信相手及び該通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶しておき、

該情報をもとに、前記更新手順を、

それぞれの前記サーバに、該サーバが前記相互認証に使用するための、更新用の新証明鍵を用いて正当性を確認可能な新デジタル証明書である前記新サーバ証明書を送信してこれを記憶させる手順を、該サーバの通信相手となる全てのクライアントから前記新証明鍵を記憶した旨の応答があった後に行うよう定めることを特徴とする更新手順決定方法。

【請求項２３】　クライアント・サーバシステムを構成する１又は複数のクライアント及び１又は複数のサーバと通信可能なデジタル証明書管理装置を制御

するコンピュータを、

前記各クライアントと前記各サーバとの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、

前記クライアント・サーバシステムを構成する各ノードについて、該ノードの通信相手及び該通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶する構成記憶手段と、

該構成記憶手段に記憶している情報をもとに、前記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段として機能させるためのプログラムであって、

前記証明鍵更新手段は、

更新用の新証明鍵を取得する手段と、

該新証明鍵を用いて正当性を確認可能な、前記相互認証に使用するための新デジタル証明書を取得する手段と、

前記クライアントのための新デジタル証明書である新クライアント証明書と、前記新証明鍵とをそれぞれ前記クライアントに送信してこれを記憶するよう要求する第１の更新要求手段と、

前記サーバのための新デジタル証明書である新サーバ証明書と、前記新証明鍵とをそれぞれ前記サーバに送信してこれを記憶するよう要求する第２の更新要求手段との機能を有し、

前記更新順制御手段が、前記第２の更新要求手段がそれぞれの前記サーバに対して前記新サーバ証明書を送信してこれを記憶するよう要求する動作を、該サーバの通信相手となる全てのクライアントからの前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御するようにしたことを特徴とするプログラム。

【請求項２４】　請求項２３記載のプログラムであって、

前記更新順制御手段が、前記第１の更新要求手段がそれぞれの前記クライアントに前記新クライアント証明書を送信してこれを記憶するよう要求する動作を、該クライアントの通信相手となる全てのサーバからの前記新証明鍵を記憶した旨の応答があった後に行うように前記更新手順を制御するようにしたことを特徴と

するプログラム。

【請求項２５】　請求項２３記載のプログラムであって、

前記更新順制御手段が、前記第１の更新要求手段が前記新クライアント証明書と前記新証明鍵とを同時に前記各クライアントに送信してこれらを記憶するよう要求し、前記第２の更新要求手段が、それぞれの前記サーバに対して、該サーバの通信相手となる全てのクライアントからの前記新証明鍵を記憶した旨の応答があった後で、前記新サーバ証明書と前記新証明鍵とを同時に前記サーバに送信し、これらを記憶するよう要求するように前記更新手順を制御するようにしたことを特徴とするプログラム。

【請求項２６】　請求項２３乃至２５のいずれか一項記載のプログラムであって、

前記コンピュータを、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む証明鍵証明書を取得する手段として機能させるためのプログラムをさらに含み、

前記第１の更新要求手段が、前記新証明鍵を前記証明鍵証明書の形式で前記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求するようにし、

前記第２の更新要求手段が、前記新証明鍵を前記証明鍵証明書の形式で前記各サーバに送信してここに含まれる証明鍵を記憶するよう要求するようにしたことを特徴とするプログラム。

【請求項２７】　請求項２３乃至２５のいずれか一項記載のプログラムであって、

前記コンピュータを、

従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第１の証明鍵証明書を取得する手段と、

前記新証明鍵を用いて正当性を確認可能なデジタル証明書であって前記新証明鍵を含む第２の証明鍵証明書を取得する手段として機能させるためのプログラムをさらに含み、

前記第１の更新要求手段が、前記新証明鍵を前記第１及び第２の証明鍵証明書

の形式でそれぞれ前記各クライアントに送信してこれを記憶するよう要求し、前記各クライアントに、前記第２の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び前記第１の証明鍵証明書を削除させる機能を有し、

前記第２の更新要求手段が、前記新証明鍵を前記第１及び第２の証明鍵証明書の形式でそれぞれ前記各サーバに送信してこれを記憶するよう要求し、前記各サーバに、前記第２の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び前記第１の証明鍵証明書を削除させる機能を有し、

前記更新順制御手段が、前記第１の更新要求手段が前記第２の証明鍵証明書をそれぞれの前記クライアントに送信してこれを記憶するよう要求する動作を、少なくとも該クライアントの通信相手となる全てのサーバから前記新サーバ証明書を記憶した旨の応答があった後に行い、前記第２の更新要求手段が前記第２の証明鍵証明書をそれぞれの前記サーバに送信してこれを記憶するよう要求する動作を、少なくとも該サーバの通信相手となる全てのクライアントから前記新クライアント証明書を記憶した旨の応答があった後に行うように前記更新手順を制御するようにしたことを特徴とするプログラム。

【請求項２８】　請求項２３乃至２７のいずれか一項記載のプログラムであって、

前記相互認証は、ＳＳＬ又はＴＬＳのプロトコルに従った相互認証であり、

前記クライアント証明書及び前記サーバ証明書はそれぞれ前記各クライアント及び前記各サーバの公開鍵証明書であることを特徴とするプログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

この発明は、デジタル証明書管理装置によってクライアント・サーバシステムを構成する１又は複数のクライアントと１又は複数のサーバの間の認証処理に用いるデジタル証明書を管理するデジタル証明書管理システム、このようなシステムを構成するデジタル証明書管理装置、このようにデジタル証明書を管理するデジタル証明書管理方法、このデジタル証明書の管理に際してそのデジタル証明書の正当性を確認するための証明鍵を更新する場合の更新手順決定方法、およびコ

ンピュータを上記のデジタル証明書管理装置として機能させるためのプログラムに関する。

【０００２】

【従来の技術】

従来から、ＰＣ等のコンピュータを複数台ネットワークを介して通信可能に接続し、少なくとも１台をサーバ装置（サーバ）、別の少なくとも１台をクライアント装置（クライアント）としたクライアント・サーバシステムを構成することが行われている。

このようなクライアント・サーバシステムにおいては、クライアント装置からサーバ装置に要求を送信し、サーバ装置がその要求に従った処理を行ってクライアント装置に対して応答を返す。そして、このようなクライアント・サーバシステムは、クライアント装置から商品の注文要求を送信し、サーバ装置においてその注文を受け付けるといった、いわゆる電子商取引にも広く用いられるようになっている。また、種々の電子装置にクライアント装置あるいはサーバ装置の機能を持たせてネットワークを介して接続し、相互間の通信によって電子装置の遠隔管理を行うシステムも提案されている。

【０００３】

このような場合においては、通信相手が適切か、あるいは送信される情報が改竄されていないかといった確認が重要である。また、特にインターネットにおいては、情報が通信相手に到達するまでに無関係なコンピュータを経由する場合が多いことから、機密情報を送信する場合、その内容を盗み見られないようにする必要もある。そして、このような要求に応える通信プロトコルとして、例えばＳＳＬ（Secure Socket Layer）と呼ばれるプロトコルが開発されており、広く用いられている。このプロトコルを用いて通信を行うことにより、公開鍵暗号方式と共通鍵暗号方式とを組み合わせ、通信相手の認証を行うと共に、情報の暗号化により改竄及び盗聴の防止を図ることができる。

【０００４】

ここで、このＳＳＬを用いて相互認証を行う場合の通信手順について、認証処理の部分に焦点を当てて説明する。図４６は、クライアント装置とサーバ装置と

がＳＳＬによる相互認証を行う際の各装置において実行する処理のフローチャートを、その処理に用いる情報と共に示す図である。

図４６に示すように、ＳＳＬによる相互認証を行う際には、まずクライアント装置側にルート鍵証明書，クライアント私有鍵，クライアント公開鍵証明書（クライアント証明書）を記憶させておく必要がある。クライアント私有鍵は、認証局（ＣＡ：certificate authority）がクライアント装置に対して発行した私有鍵である。そして、クライアント公開鍵証明書は、その私有鍵と対応する公開鍵にＣＡがデジタル署名を付してデジタル証明書としたものである。また、ルート鍵証明書は、ＣＡがデジタル署名に用いた証明用私有鍵であるルート私有鍵と対応する証明用公開鍵（以下「証明鍵」ともいう）であるルート鍵に、デジタル署名を付してデジタル証明書としたものである。

【０００５】

図４７にこれらの関係を示す。

図４７（ａ）に示すように、クライアント公開鍵は、クライアント私有鍵を用いて暗号化された文書を復号化するための鍵本体と、その公開鍵の発行者（ＣＡ），発行相手（クライアント装置），有効期限等の情報を含む書誌情報とによって構成される。そして、ＣＡは、鍵本体や書誌情報が改竄されていないことを示すため、クライアント公開鍵をハッシュ処理して得たハッシュ値を、ルート私有鍵を用いて暗号化し、デジタル署名としてクライアント公開鍵に付す。またこの際に、デジタル署名に用いるルート私有鍵の識別情報を署名鍵情報として公開鍵の書誌情報に加える。そして、このデジタル署名を付した公開鍵証明書が、クライアント公開鍵証明書である。

【０００６】

このクライアント公開鍵証明書を認証処理に用いる場合には、ここに含まれるデジタル署名を、ルート私有鍵と対応する公開鍵であるルート鍵の鍵本体を用いて復号化する。この復号化が正常に行われれば、デジタル署名が確かにＣＡによって付されたことがわかる。また、クライアント公開鍵部分をハッシュ処理して得たハッシュ値と、復号して得たハッシュ値とが一致すれば、鍵自体も損傷や改竄を受けていないことがわかる。さらに、受信したデータをこのクライアント公

開鍵を用いて正常に復号化できれば、そのデータは、クライアント私有鍵の持ち主、つまりクライアント装置から送信されたものであることがわかる。あとは、書誌情報を参照して、ＣＡの信頼性やクライアント装置の登録有無等によって認証の正否を決定すればよい。

【０００７】

ここで、認証を行うためには、ルート鍵を予め記憶しておく必要があるが、このルート鍵も、図４７（ｂ）に示すように、ＣＡがデジタル署名を付したルート鍵証明書として記憶しておく。このルート鍵証明書は、自身に含まれる公開鍵でデジタル署名を復号化可能な、自己署名形式である。そして、ルート鍵を使用する際に、そのルート鍵証明書に含まれる鍵本体を用いてデジタル署名を復号化し、ルート鍵をハッシュ処理して得たハッシュ値と比較する。これが一致すれば、ルート鍵が破損等していないことを確認できるのである。

【０００８】

図４６の説明に戻ると、サーバ装置側には、ルート鍵証明書，サーバ私有鍵，サーバ公開鍵証明書（サーバ証明書）を記憶させておく必要がある。サーバ私有鍵及びサーバ公開鍵証明書は、ＣＡがサーバ装置に対して発行した私有鍵及び公開鍵証明書である。ここではクライアント装置とサーバ装置に対して同じＣＡが同じルート私有鍵を用いて証明書を発行しているものとし、この場合にはルート鍵証明書はクライアント装置とサーバ装置で共通となる。

【０００９】

フローチャートの説明に入る。なお、図４６において、２本のフローチャート間の矢印は、データの転送を示し、送信側は矢印の根元のステップで転送処理を行い、受信側はその情報を受信すると矢印の先端のステップの処理を行うものとする。また、各ステップの処理が正常に完了しなかった場合には、その時点で認証失敗の応答を返して処理を中断するものとする。相手から認証失敗の応答を受けた場合、処理がタイムアウトした場合等も同様である。

【００１０】

クライアント・サーバシステムにおいて、接続を要求するのはクライアント装置側であるが、ユーザの指示等によってこの必要が生じた場合、クライアント装

置のＣＰＵは、所要の制御プログラムを実行することにより、図４６の左側に示すフローチャートの処理を開始する。そして、ステップＳ１１でサーバ装置に対して接続要求を送信する。

一方サーバ装置のＣＰＵは、この接続要求を受信すると、所要の制御プログラムを実行することにより、図４６の右側に示すフローチャートの処理を開始する。そして、ステップＳ２１で第１の乱数を生成し、これをサーバ私有鍵を用いて暗号化する。そして、ステップＳ２２でその暗号化した第１の乱数とサーバ公開鍵証明書とをクライアント装置に送信する。このステップＳ２２の処理において、サーバ装置のＣＰＵが第１のサーバ側認証処理手段として機能する。

【００１１】

クライアント装置側では、これを受信すると、ステップＳ１２でルート鍵証明書を用いてサーバ公開鍵証明書の正当性を確認する。これには、上述のように損傷や改竄を受けていないことを確認するのみならず、書誌情報を参照してサーバ装置が適当な通信相手であることを確認する処理を含む。

そして確認ができると、ステップＳ１３で、受信したサーバ公開鍵証明書に含まれるサーバ公開鍵を用いて第１の乱数を復号化する。ここで復号化が成功すれば、第１の乱数は確かにサーバ公開鍵証明書の発行対象であるサーバ装置から受信したものだと確認できる。そして、サーバ装置を正当な通信相手として認証する。このステップＳ１２及びＳ１３の処理において、クライアント装置のＣＰＵが第２のクライアント側認証処理手段として機能する。

【００１２】

その後、ステップＳ１４でこれとは別に第２の乱数及び第３の乱数を生成する。そして、ステップＳ１５で第２の乱数をクライアント私有鍵を用いて暗号化し、第３の乱数をサーバ公開鍵を用いて暗号化し、ステップＳ１６でこれらをクライアント公開鍵証明書と共にサーバ装置に送信する。第３の乱数の暗号化は、サーバ装置以外の装置に乱数を知られないようにするために行うものである。このステップＳ１６の処理において、クライアント装置のＣＰＵが第１のクライアント側認証処理手段として機能する。

【００１３】

サーバ装置側では、これを受信すると、ステップＳ２３でルート鍵証明書を用いてクライアント公開鍵証明書の正当性を確認する。これにも、ステップＳ１２の場合と同様、クライアント装置が適当な通信相手であることを確認する処理を含む。そして確認ができると、ステップＳ２４で、受信したクライアント公開鍵証明書に含まれるクライアント公開鍵を用いて第２の乱数を復号化する。ここで復号化が成功すれば、第２の乱数は確かにクライアント公開鍵証明書の発行対象であるクライアント装置から受信したものだと確認できる。そして、サーバ装置を正当な通信相手として認証する。このステップＳ２３及び２４の処理において、サーバ装置のＣＰＵが第２のサーバ側認証処理手段として機能する。

その後、ステップＳ２５でサーバ私有鍵を用いて第３の乱数を復号化する。ここまでの処理で、サーバ側とクライアント側に共通の第１乃至第３の乱数が共有されたことになる。そして、少なくとも第３の乱数は、生成したクライアント装置と、サーバ私有鍵を持つサーバ装置以外の装置が知ることはない。ここまでの処理が成功すると、ステップＳ２６でクライアント装置に対して認証成功の応答を返す。

【００１４】

クライアント装置側では、これを受信すると、ステップＳ１７で第１乃至第３の乱数から共通鍵を生成し、以後の通信の暗号化に用いるものとして認証処理を終了する。サーバ装置側でも、ステップＳ２７で同様の処理を行って終了する。そして、以上の処理によって互いに通信を確立し、以後はステップＳ１７又はＳ２７で生成した共通鍵を用い、共通鍵暗号方式でデータを暗号化して通信を行う。

このような処理を行うことにより、クライアント装置とサーバ装置が互いに相手を認証した上で安全に共通鍵を交換することができ、通信を確かな相手と安全に行うことができる。

【００１５】

ところで、公開鍵暗号方式においては、鍵長にもよるが、時間をかければ公開鍵から私有鍵を導くことができる。そして、私有鍵がわかってしまえば、第３者がその私有鍵の持ち主になりすますことが可能になるので、認証の確実性や通信

の安全性が保たれない。そこで、上述のように鍵に有効期限を設け、所定期間毎に鍵のセットを更新するというセキュリティポリシーを採用するユーザが増えている。このため、例えば上記のような相互認証を利用した遠隔管理システム等を提供する場合には、顧客に対し、鍵の更新が可能なシステムであるという保証を行う必要が生じている。これは、ルート鍵とルート私有鍵についても同様である。なお、鍵の更新事由としては、所定の有効期限の到来の他にも、私有鍵の第３者への漏洩が判明した場合等が考えられる。

このような鍵の更新に関する技術としては、例えば特許文献１に記載のものが挙げられる。

【００１６】

【特許文献１】

特開平１１－１２２２３８号公報

【００１７】

【発明が解決しようとする課題】

しかしながら、特許文献１には、各装置に対して発行した鍵の更新に関する記載はあるが、ルート鍵の更新についての記載はない。

公開鍵暗号方式の場合、各装置に発行した鍵のペアを更新する場合には、その装置には新たな私有鍵に対応した新たな公開鍵証明書が記憶されることになり、通信相手にこれを渡せば、図４６に示した認証処理を支障なく行うことができる。

しかし、ルート鍵を更新する場合、新たなルート鍵では従前のデジタル証明書に付されたデジタル署名を復号化することができないため、新たなルート鍵と対応する新たなルート私有鍵を用いて各装置の公開鍵証明書を作成し直し、これを配布しなければ、図４６に示した認証処理の実行に支障を来してしまう（ただし、各装置の私有鍵は必ずしも更新する必要はない）。

【００１８】

そして、認証処理に支障を来さずにルート鍵を更新する方式が知られていなかったため、更新の必要な装置にルート鍵をネットワークを介して安全に送信することができなかった。そこで、ルート鍵証明書や新たな公開鍵証明書を別の安全

な経路で各装置に届ける必要があったのである。

この経路としては、例えば書留郵便が考えられ、証明書のデータを記録したメモリカードやフレキシブルディスク等の記録媒体を装置の管理者に書留郵便で送付し、管理者が装置の鍵を更新するという方式が考えられる。しかし、この方式では、クライアントやサーバの各装置について十分な知識を持った管理者がいる場合にしか適用できないし、ＣＡ側は記録媒体を送付した後の処理については装置の管理者を信用するしかなかった。従って、管理者が更新処理を怠ったり誤ったりした場合には、認証処理が行えなくなってしまうという問題があった。

【００１９】

一方管理者側も、受け取った証明書が正しいものであるか否かは、封筒やデータに記載された送り主の名称等を信用して判断するしかなく、ＣＡの名を騙る別人から受け取ったニセの証明書を装置に記憶させてしまうといった危険は常につきまとうことになる。

また、ＣＡやクライアント・サーバシステムによるサービスの提供者が、各装置の配置先にサービスマンを派遣して鍵の更新を行うことも考えられるが、広い地域でこのような方式を採るには多数のサービス拠点が必要になり、コストが嵩むことになる。また、サービスマンの教育や不正防止、更新作業用の管理者ＩＤの管理も問題となる。例えば、認証情報を手入力する単純な方式を採ろうとすると、退職したサービスマンについての更新権限を抹消するためには、各装置に記憶させている認証情報を変更する必要があるが、顧客先に設置された多数の装置にこのような変更を行うことは困難である。

【００２０】

結局のところ、ネットワークを介さずに証明書の安全な配布経路を確保するためには、人間を信用する他なく、そこには欺瞞が入りこむ余地が出てしまう。そして、この余地を小さくするよう管理することはできるが、そのためには膨大なコストが生じてしまい、欺瞞の危険を考慮しなくて済むレベルの経路を証明書の配布のために構築することは、現実的ではなかった。

従来の技術の項で述べたとおり、ＳＳＬを用いた相互認証は理論的には可能なのであるが、以上のような問題があり、ルート鍵の安全な更新が実質的に不可能

であったので、実際には使用されていないのが現状である。またこのような問題は、公開鍵暗号とデジタル証明書を用いて認証を行う他のプロトコルを用いた場合にも、同様に発生するものと考えられる。

【００２１】

この発明は、このような問題を解決し、クライアント・サーバシステムにおける認証処理でデジタル証明書の正当性を確認するために用いる証明鍵を、自動的に更新できるようにすることを目的とする。そして、このことにより、公開鍵暗号を利用したデジタル証明書を用いるＳＳＬ等の方式による相互認証を、クライアント・サーバシステムにおいて低コストで実現可能とすることを目的とする。

【００２２】

【課題を解決するための手段】

上記の目的を達成するため、この発明のデジタル証明書管理装置は、１又は複数のクライアントと１又は複数のサーバとによって構成され、その各クライアントと各サーバとの間でデジタル証明書を用いて相互認証を行うようにしたクライアント・サーバシステムに、上記各クライアント及び上記各サーバと通信可能なデジタル証明書管理装置を接続したデジタル証明書管理システムにおいて、上記デジタル証明書管理装置に、上記各クライアント及び上記各サーバが上記相互認証に使用する上記デジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、上記クライアント・サーバシステムを構成する各ノードについて、そのノードの通信相手及びその通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶する構成記憶手段と、その構成記憶手段に記憶している情報をもとに、上記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段とを設け、上記証明鍵更新手段に、更新用の新証明鍵を取得する手段と、その新証明鍵を用いて正当性を確認可能な、上記相互認証に使用するための新デジタル証明書を取得する手段と、上記各クライアントのための新デジタル証明書である新クライアント証明書と、上記新証明鍵とをそれぞれ対応する上記各クライアントに送信してこれを記憶するよう要求する第１の更新要求手段と、上記各サーバのための新デジタル証明書である新サーバ証明書と、上記新証明鍵とをそれぞれ対応する上記各サーバに送信してこれを記憶するよう要求する

第２の更新要求手段とを設け、上記更新順制御手段を、上記第２の更新要求手段がそれぞれの上記サーバに上記新サーバ証明書を送信してこれを記憶するよう要求する動作を、そのサーバの通信相手となる全てのクライアントから上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御する手段としたものである。

【００２３】

このようなデジタル証明書管理システムにおいて、上記デジタル証明書管理装置の上記更新順制御手段を、上記第１の更新要求手段がそれぞれの上記クライアントに上記新クライアント証明書を送信してこれを記憶するよう要求する動作を、そのクライアントの通信相手となる全てのサーバから上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御する手段とするとよい。

あるいは、上記デジタル証明書管理装置の上記更新順制御手段を、上記第１の更新要求手段が上記新クライアント証明書と上記新証明鍵とを同時に上記各クライアントに送信してこれらを記憶するよう要求し、上記第２の更新要求手段が、それぞれの上記サーバに対して、そのサーバの通信相手となる全てのクライアントから上記新証明鍵を記憶した旨の応答があった後で、上記新サーバ証明書と上記新証明鍵とを同時に送信し、これらを記憶するよう要求するように上記更新手順を制御する手段とするとよい。

【００２４】

これらのデジタル証明書管理システムにおいて、上記各サーバに、上記デジタル証明書管理装置と少なくとも一つの上記クライアントとの間の通信を仲介する手段を設け、上記デジタル証明書管理装置と上記各クライアントとはいずれかの上記サーバを介して通信を行うようにするとよい。

さらに、上記各クライアントに、上記デジタル証明書管理装置とそのクライアントとの間の通信を仲介するサーバに対して定期的に通信を要求する手段を設け、上記サーバから上記クライアントへ送信すべき情報は、その通信の要求に対する応答として送信するようにするとよい。

あるいは、上記のこれらのデジタル証明書管理システムにおいて、上記各クライアントに、上記デジタル証明書管理装置と少なくとも一つの上記サーバとの間

の通信を仲介する手段を設け、上記デジタル証明書管理装置と上記各サーバとはいずれかの上記クライアントを介して通信を行うようにするとよい。

【００２５】

さらに、これらのデジタル証明書管理システムにおいて、上記デジタル証明書管理装置の上記証明鍵更新手段に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む証明鍵証明書を取得する手段を設け、上記第１の更新要求手段を、上記新証明鍵を上記証明鍵証明書の形式で上記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求する手段とし、上記第２の更新要求手段を、上記新証明鍵を上記証明鍵証明書の形式で上記各サーバに送信してここに含まれる証明鍵を記憶するよう要求する手段とし、上記各クライアント及び上記各サーバにそれぞれ、上記デジタル証明書管理装置から上記証明鍵証明書に含まれる証明鍵の記憶を要求された場合に、受信した証明鍵証明書の正当性を従前の証明鍵を用いて確認し、そこに含まれる証明鍵が適当なものであると判断した場合にその証明鍵を記憶する手段を設けるとよい。

【００２６】

あるいは、上記デジタル証明書管理装置の上記証明鍵更新手段に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第１の証明鍵証明書を取得する手段と、上記新証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第２の証明鍵証明書を取得する手段とを設け、上記第１の更新要求手段を、上記新証明鍵を上記第１及び第２の証明鍵証明書の形式でそれぞれ上記各クライアントに送信してこれを記憶するよう要求する手段とし、上記第２の更新要求手段を、上記新証明鍵を上記第１及び第２の証明鍵証明書の形式でそれぞれ上記各サーバに送信してこれを記憶するよう要求する手段とし、上記各クライアント及び上記各サーバにそれぞれ、上記デジタル証明書管理装置から上記第１の証明鍵証明書を記憶するよう要求された場合に、その証明書の正当性を従前の証明鍵を用いて確認し、これが適当なものであると判断した場合にその証明書を記憶する手段と、上記デジタル証明書管理装置から上記第２の証明鍵証明書を記憶するよう要求された場合に、その証明書の正当性を上記第１の証明鍵証明書に含まれる上記新証明鍵を用いて確認し、上記第２の

証明鍵証明書が適当なものであると判断した場合に、その証明書を記憶すると共に従前の証明鍵証明書及び上記第１の証明鍵証明書を削除する手段とを設け、上記デジタル証明書管理装置の上記更新順制御手段を、上記第１の更新要求手段が上記第２の証明鍵証明書をそれぞれの上記クライアントに送信してこれを記憶するよう要求する動作を、少なくともそのクライアントの通信相手となる全てのサーバから上記新サーバ証明書を記憶した旨の応答があった後に行うよう制御し、上記第２の更新要求手段が上記第２の証明鍵証明書をそれぞれの上記サーバに送信してこれを記憶するよう要求する動作を、少なくともそのサーバの通信相手となる全てのクライアントから上記新クライアント証明書を記憶した旨の応答があった後に行うように上記更新手順を制御する手段とするとよい。

【００２７】

さらに、上記クライアントと上記サーバが行う上記相互認証を、ＳＳＬ又はＴＬＳのプロトコルに従った相互認証とし、上記クライアント証明書及び上記サーバ証明書をそれぞれ上記各クライアント及び上記各サーバの公開鍵証明書とするとよい。

【００２８】

また、この発明のデジタル証明書管理装置は、クライアント・サーバシステムを構成する１又は複数のクライアント及び１又は複数のサーバと通信可能なデジタル証明書管理装置において、上記各クライアントと上記各サーバとの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、上記クライアント・サーバシステムを構成する各ノードについて、そのノードの通信相手及びその通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶する構成記憶手段と、その構成記憶手段に記憶している情報をもとに、上記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段とを設け、上記証明鍵更新手段に、更新用の新証明鍵を取得する手段と、その新証明鍵を用いて正当性を確認可能な、上記相互認証に使用するための新デジタル証明書を取得する手段と、上記クライアントのための新デジタル証明書である新クライアント証明書と、上記新証明鍵とをそれぞれ上記クライアントに送信してこれを記憶するよう要求する第１の更新要求手段と、上記サーバ

のための新デジタル証明書である新サーバ証明書と、上記新証明鍵とをそれぞれ上記サーバに送信してこれを記憶するよう要求する第２の更新要求手段とを設け、上記更新順制御手段を、上記第２の更新要求手段がそれぞれの上記サーバに対して上記新サーバ証明書を送信してこれを記憶するよう要求する動作を、そのサーバの通信相手となる全てのクライアントからの上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御する手段としたものである。

【００２９】

このようなデジタル証明書管理装置において、上記更新順制御手段を、上記第１の更新要求手段がそれぞれの上記クライアントに上記新クライアント証明書を送信してこれを記憶するよう要求する動作を、そのクライアントの通信相手となる全てのサーバからの上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御する手段とするとよい。

さらに、上記更新順制御手段を、上記第１の更新要求手段が上記新クライアント証明書と上記新証明鍵とを同時に上記各クライアントに送信してこれらを記憶するよう要求し、上記第２の更新要求手段が、それぞれの上記サーバに対して、そのサーバの通信相手となる全てのクライアントからの上記新証明鍵を記憶した旨の応答があった後で、上記新サーバ証明書と上記新証明鍵とを同時に送信し、これらを記憶するよう要求するように上記更新手順を制御する手段とするとよい。

【００３０】

これらのデジタル証明書管理装置において、上記証明鍵更新手段に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む証明鍵証明書を取得する手段を設け、上記第１の更新要求手段を、上記新証明鍵を上記証明鍵証明書の形式で上記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求する手段とし、上記第２の更新要求手段を、上記新証明鍵を上記証明鍵証明書の形式で上記各サーバに送信してここに含まれる証明鍵を記憶するよう要求する手段とするとよい。

【００３１】

あるいは、上記証明鍵更新手段に、従前の証明鍵を用いて正当性を確認可能な

デジタル証明書であって上記新証明鍵を含む第１の証明鍵証明書を取得する手段と、上記新証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第２の証明鍵証明書を取得する手段とを設け、上記第１の更新要求手段が、上記新証明鍵を上記第１及び第２の証明鍵証明書の形式でそれぞれ上記各クライアントに送信してこれを記憶するよう要求する手段であって、上記各クライアントに、上記第２の証明鍵証明書を記憶する場合に従前の証明鍵証明書及び上記第１の証明鍵証明書を削除させる手段を有するようにし、上記第２の更新要求手段を、上記新証明鍵を上記第１及び第２の証明鍵証明書の形式でそれぞれ上記サーバに送信してこれを記憶するよう要求する手段であって、上記サーバに、上記第２の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び上記第１の証明鍵証明書を削除させる手段を有するようにし、上記更新順制御手段を、上記第１の更新要求手段が上記第２の証明鍵証明書をそれぞれの上記クライアントに送信してこれを記憶するよう要求する動作を、少なくともそのクライアントの通信相手となる全てのサーバから上記新サーバ証明書を記憶した旨の応答があった後に行い、上記第２の更新要求手段が上記第２の証明鍵証明書をそれぞれの上記サーバに送信してこれを記憶するよう要求する動作を、少なくともそのサーバの通信相手となる全てのクライアントから上記新クライアント証明書を記憶した旨の応答があった後に行うように上記更新手順を制御する手段とするとよい。

【００３２】

これらのデジタル証明書管理装置において、上記相互認証を、ＳＳＬ又はＴＬＳのプロトコルに従った相互認証とし、上記クライアント証明書及び上記サーバ証明書をそれぞれ上記各クライアント及び上記各サーバの公開鍵証明書とするとよい。

【００３３】

また、この発明のデジタル証明書管理方法は、クライアント・サーバシステムを構成する１又は複数のクライアントと１又は複数のサーバとの間で相互認証に使用するデジタル証明書を、上記各クライアント及び上記各サーバと通信可能なデジタル証明書管理装置によって管理するデジタル証明書管理方法において、上記デジタル証明書管理装置を、上記クライアント・サーバシステムを構成する各

ノードについて、そのノードの通信相手及びその通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶しておき、その情報をもとに定める更新手順に従って上記各クライアント及び上記各サーバが上記相互認証に使用する上記デジタル証明書の正当性を確認するための証明鍵を更新し、その証明鍵の更新を、更新用の新証明鍵を取得する手順と、その新証明鍵を用いて正当性を確認可能な、上記相互認証に使用するための新デジタル証明書を取得する手順と、上記各クライアントのための新デジタル証明書である新クライアント証明書と、上記新証明鍵とをそれぞれ対応する上記各クライアントに送信してこれを記憶させる手順と、上記各サーバのための新デジタル証明書である新サーバ証明書と、上記新証明鍵とをそれぞれ対応する上記各サーバに送信してこれを記憶させる手順とを実行することによって行い、上記更新手順を、それぞれの上記サーバに上記新サーバ証明書を送信してこれを記憶させる手順をそのサーバの通信相手となる全てのクライアントから上記新証明鍵を記憶した旨の応答があった後に行うよう定めるようにするとよい。

【００３４】

このようなデジタル証明書管理方法において、上記更新手順を、それぞれの上記クライアントに上記新クライアント証明書を送信してこれを記憶させる手順をそのクライアントの通信相手となる全てのサーバから上記新証明鍵を記憶した旨の応答があった後に行うよう定めるようにするとよい。

あるいは、上記更新手順を、上記新クライアント証明書と上記新証明鍵とを同時に上記各クライアントに送信してこれらを記憶させるように定め、さらに、それぞれの上記サーバに対して、そのサーバの通信相手となる全てのクライアントから上記新証明鍵を記憶した旨の応答があった後で、上記新サーバ証明書と上記新証明鍵とを同時に送信し、これらを記憶させるように定めるようにするとよい。

【００３５】

これらのデジタル証明書管理方法において、上記証明鍵の更新の際に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む証明鍵証明書を取得する手順をさらに実行し、上記新証明鍵を上記各サーバある

いは上記各クライアントに送信してこれを記憶させる手順において、その新証明鍵を上記証明鍵証明書の形式で送信してここに含まれる証明鍵を記憶させるようにし、上記各クライアント又は上記各サーバに上記証明鍵証明書に含まれる証明鍵を記憶させる場合に、その証明鍵証明書の正当性を、記憶している従前の証明鍵を用いて確認させ、そこに含まれる証明鍵が適当なものであると判断した場合にその証明鍵を記憶させるようにするとよい。

【００３６】

あるいは、上記証明鍵の更新の際に、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第１の証明鍵証明書を取得する手順と、上記新証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第２の証明鍵証明書を取得する手順とをさらに実行し、上記新証明鍵を上記各サーバあるいは上記各クライアントに送信してこれを記憶させる手順において、その新証明鍵を上記第１及び第２の証明鍵証明書の形式でそれぞれ送信してこれを記憶させるようにし、上記更新手順を、上記第２の証明鍵証明書をそれぞれの上記クライアントに送信してこれを記憶するよう要求する手順を少なくともそのクライアントの通信相手となる全てのサーバから上記新サーバ証明書を記憶した旨の応答があった後に行うよう定め、さらに、上記第２の証明鍵証明書をそれぞれの上記サーバに送信してこれを記憶するよう要求する動作を少なくともそのサーバの通信相手となる全てのクライアントから上記新クライアント証明書を記憶した旨の応答があった後に行うよう定め、上記各クライアント又は上記各サーバに上記第１の証明鍵証明書を記憶させる際に、その証明書の正当性を従前の証明鍵を用いて確認させ、これが適当なものであると判断した場合にその証明書を記憶させ、上記各クライアント又は上記各サーバに上記第２の証明鍵証明書を記憶させる際に、その証明書の正当性を上記第１の証明鍵証明書に含まれる上記新証明鍵を用いて確認させ、上記第２の証明鍵証明書が適当なものであると判断した場合に、その証明書を記憶させると共に従前の証明鍵証明書及び上記第１の証明鍵証明書を削除させるようにするとよい。

【００３７】

さらに、これらのデジタル証明書管理方法において、上記クライアントと上記

サーバとの間の上記相互認証を、ＳＳＬ又はＴＬＳのプロトコルに従った相互認証とし、上記クライアント証明書及び上記サーバ証明書をそれぞれ上記各クライアント及び上記各サーバの公開鍵証明書とするとよい。

【００３８】

また、この発明の更新手順決定方法は、クライアント・サーバシステムを構成する１又は複数のクライアントと１又は複数のサーバとに記憶させ、これらの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を、上記各クライアント及び上記各サーバと通信可能なデジタル証明書管理装置によって更新する際の更新手順を定める更新手順決定方法であって、上記デジタル証明書管理装置が、上記クライアント・サーバシステムを構成する各ノードについて、そのノードの通信相手及びその通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶しておき、その情報をもとに、上記更新手順を、それぞれの上記サーバに、そのサーバが上記相互認証に使用するための、更新用の新証明鍵を用いて正当性を確認可能な新デジタル証明書である上記新サーバ証明書を送信してこれを記憶させる手順を、そのサーバの通信相手となる全てのクライアントから上記新証明鍵を記憶した旨の応答があった後に行うよう定めるものである。

【００３９】

また、この発明のプログラムは、クライアント・サーバシステムを構成する１又は複数のクライアント及び１又は複数のサーバと通信可能なデジタル証明書管理装置を制御するコンピュータを、上記各クライアントと上記各サーバとの間で相互認証に使用するデジタル証明書の正当性を確認するための証明鍵を更新する証明鍵更新手段と、上記クライアント・サーバシステムを構成する各ノードについて、そのノードの通信相手及びその通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶する構成記憶手段と、その構成記憶手段に記憶している情報をもとに、上記証明鍵更新手段による証明鍵の更新手順を制御する更新順制御手段として機能させるためのプログラムにおいて、上記証明鍵更新手段に、更新用の新証明鍵を取得する手段と、その新証明鍵を用いて正当性を確認可能な、上記相互認証に使用するための新デジタル証明書を取得する手段と

、上記クライアントのための新デジタル証明書である新クライアント証明書と、上記新証明鍵とをそれぞれ上記クライアントに送信してこれを記憶するよう要求する第１の更新要求手段と、上記サーバのための新デジタル証明書である新サーバ証明書と、上記新証明鍵とをそれぞれ上記サーバに送信してこれを記憶するよう要求する第２の更新要求手段との機能を設け、上記更新順制御手段が、上記第２の更新要求手段がそれぞれの上記サーバに対して上記新サーバ証明書を送信してこれを記憶するよう要求する動作を、そのサーバの通信相手となる全てのクライアントからの上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御するようにしたものである。

【００４０】

このようなプログラムにおいて、上記更新順制御手段が、上記第１の更新要求手段がそれぞれの上記クライアントに上記新クライアント証明書を送信してこれを記憶するよう要求する動作を、そのクライアントの通信相手となる全てのサーバからの上記新証明鍵を記憶した旨の応答があった後に行うように上記更新手順を制御するようにするとよい。

さらに、上記更新順制御手段が、上記第１の更新要求手段が上記新クライアント証明書と上記新証明鍵とを同時に上記各クライアントに送信してこれらを記憶するよう要求し、上記第２の更新要求手段が、それぞれの上記サーバに対して、そのサーバの通信相手となる全てのクライアントからの上記新証明鍵を記憶した旨の応答があった後で、上記新サーバ証明書と上記新証明鍵とを同時に上記サーバに送信し、これらを記憶するよう要求するように上記更新手順を制御するようにするとよい。

【００４１】

また、これらのプログラムにおいて、上記コンピュータを、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む証明鍵証明書を取得する手段として機能させるためのプログラムをさらに含め、上記第１の更新要求手段が、上記新証明鍵を上記証明鍵証明書の形式で上記各クライアントに送信してここに含まれる証明鍵を記憶するよう要求するようにし、上記第２の更新要求手段が、上記新証明鍵を上記証明鍵証明書の形式で上記各サーバに送信

してここに含まれる証明鍵を記憶するよう要求するようにするとよい。

【００４２】

あるいは、上記コンピュータを、従前の証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第１の証明鍵証明書を取得する手段と、上記新証明鍵を用いて正当性を確認可能なデジタル証明書であって上記新証明鍵を含む第２の証明鍵証明書を取得する手段として機能させるためのプログラムをさらに含め、上記第１の更新要求手段を、上記新証明鍵を上記第１及び第２の証明鍵証明書の形式でそれぞれ上記各クライアントに送信してこれを記憶するよう要求し、上記各クライアントに、上記第２の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び上記第１の証明鍵証明書を削除させる機能を有するようにし、上記第２の更新要求手段を、上記新証明鍵を上記第１及び第２の証明鍵証明書の形式でそれぞれ上記各サーバに送信してこれを記憶するよう要求し、上記各サーバに、上記第２の証明鍵証明書を記憶する場合には従前の証明鍵証明書及び上記第１の証明鍵証明書を削除させる機能を有するようにし、上記更新順制御手段が、上記第１の更新要求手段が上記第２の証明鍵証明書をそれぞれの上記クライアントに送信してこれを記憶するよう要求する動作を、少なくともそのクライアントの通信相手となる全てのサーバから上記新サーバ証明書を記憶した旨の応答があった後に行い、上記第２の更新要求手段が上記第２の証明鍵証明書をそれぞれの上記サーバに送信してこれを記憶するよう要求する動作を、少なくともそのサーバの通信相手となる全てのクライアントから上記新クライアント証明書を記憶した旨の応答があった後に行うように上記更新手順を制御するようにするとよい。

さらに、これらのプログラムにおいて、上記相互認証を、ＳＳＬ又はＴＬＳのプロトコルに従った相互認証とし、上記クライアント証明書及び上記サーバ証明書をそれぞれ上記各クライアント及び上記各サーバの公開鍵証明書とするとよい。

【００４３】

【発明の実施の形態】

以下、この発明の好ましい実施の形態を図面を参照して説明する。

〔第１の実施形態：図１乃至図１１〕

まず、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント及びサーバによって構成される、この発明のデジタル証明書管理システムの第１の実施形態の構成について説明する。この実施形態においては、各１つのクライアント及びサーバによってクライアント・サーバシステムを構成しており、この実施形態は、この発明を最も基本的なシステムに適用した例である。図２に、このデジタル証明書管理システムを構成する各装置の、この発明の特徴となる部分の機能構成を示す機能ブロック図を示す。図２において、この発明の特徴と関連しない部分の図示は省略している。

【００４４】

図２に示すように、このデジタル証明書管理システムは、証明書管理装置１０，サーバ装置３０，クライアント装置４０によって構成される。

そして、クライアント装置（クライアント）４０及びサーバ装置（サーバ）３０は、公開鍵暗号とデジタル証明書を用いる認証方式であるＳＳＬによる相互認証によって互いを正当な通信相手として認証した場合に、互いに通信を確立させるようにしている。そして、クライアント装置４０が送信した要求に対し、サーバ装置３０が必要な処理を行って応答を返すことにより、クライアント・サーバシステムとして機能する。証明書管理装置１０は、その相互認証に用いるデジタル証明書を発行し、またそのデジタル証明書の管理や更新等を行うための装置であり、ＣＡに相当する。

【００４５】

なお、実際のシステムにおいては、サーバ装置３０がクライアントの機能を併せ持ったり、クライアント装置４０がサーバの機能を併せ持ったりすることも考えられる。そして、サーバ装置３０がクライアントとして機能して、サーバとして機能するクライアント装置４０に要求を送信することもありうるが、このような場合には、後述する第２の実施形態に準ずる動作を行うようにすればよい。従って、ここでは後述するルート鍵更新処理においてサーバとして機能する装置をサーバ装置、クライアントとして機能する装置をクライアント装置と呼ぶものと

する。

【００４６】

このようなデジタル証明書管理システムにおいて、上述のクライアント装置４０からサーバ装置３０への送信も含め、証明書管理装置１０，サーバ装置３０，クライアント装置４０の各ノードは、ＲＰＣ（remote procedure call）により、相互の実装するアプリケーションプログラムのメソッドに対する処理の依頼である「要求」を送信し、この依頼された処理の結果である「応答」を取得することができるようになっている。

【００４７】

すなわち、サーバ装置３０又はクライアント装置４０では、証明書管理装置１０への要求を生成してこれを証明書管理装置１０へ引き渡し、この要求に対する応答を取得できる一方で、証明書管理装置１０は、クライアント・サーバシステム側への要求を生成してこれをサーバ装置３０へ引き渡し、この要求に対する応答を取得できるようになっている。この要求には、サーバ装置３０にクライアント装置４０に対して各種要求を送信させ、クライアント装置４０からの応答をサーバ装置３０を介して取得することも含まれる。

なお、ＲＰＣを実現するために、ＳＯＡＰ（Simple Object Access Protocol），ＨＴＴＰ（Hyper Text Transfer Protocol），ＦＴＰ（File Transfer Protocol），ＣＯＭ（Component Object Model），ＣＯＲＢＡ（Common Object Request Broker Architecture）等の既知のプロトコル（通信規格），技術，仕様などを利用することができる。

【００４８】

この送受信のデータ送受モデルを図３の概念図に示す。

（Ａ）は、証明書管理装置１０でクライアント装置４０に対する要求が発生したケースである。このケースでは、証明書管理装置１０が管理装置側要求ａを生成し、これをサーバ装置３０を経由して受け取ったクライアント装置４０がこの要求に対する応答ａを返すというモデルになる。なお、（Ａ）では、応答ａだけでなく応答遅延通知ａ´を返信するケースが表記されている。これは、クライアント装置４０が、サーバ装置３０を経由して管理装置側要求ａを受け取って、当

該要求に対する応答を即座に返せないと判断したときには、応答遅延通知を通知して一旦接続状態を切断し、次回の接続の際に上記要求に対する応答を改めて引き渡す構成としているためである。

なおここでは、サーバ装置３０からクライアント装置４０に対して通信を要求することはできないので、サーバ装置３０からクライアント装置４０に対して送信すべき要求は、クライアント装置４０からサーバ装置３０に対して接続要求があった場合に、これに対する応答として送信することになる。

【００４９】

（Ｂ）は、クライアント装置４０で証明書管理装置１０に対する要求が発生したケースである。このケースでは、クライアント装置４０がクライアント装置側要求ｂを生成し、これをサーバ装置３０を経由して受け取った証明書管理装置１０が、当該要求に対する応答ｂを返すというモデルになっている。なお、（Ｂ）のケースでも、応答を即座に返せないときに応答遅延通知ｂ′を返すことは（Ａ）のケースと同様である。

【００５０】

次に、このデジタル証明書管理システムを構成する各装置の構成と機能についてより詳細に説明する。

図１は、図２に示した証明書管理装置のハードウェア構成を示すブロック図である。この図に示す通り、証明書管理装置１０は、ＣＰＵ１１，ＲＯＭ１２，ＲＡＭ１３，ＨＤＤ１４，通信インタフェース（Ｉ／Ｆ）１５を備え、これらがシステムバス１６によって接続されている。そして、ＣＰＵ１１がＲＯＭ１２やＨＤＤ１４に記憶している各種制御プログラムを実行することによってこの証明書管理装置１０の動作を制御し、後述するようにこの発明に係る各手段（証明鍵更新手段，構成記憶手段，更新順制御手段，第１の更新要求手段，第２の更新要求手段，その他の手段）として機能させる。

なお、証明書管理装置１０のハードウェアとしては、適宜公知のコンピュータを採用することができる。もちろん、必要に応じて他のハードウェアを付加してもよい。

【００５１】

クライアント・サーバシステムを構成するクライアント装置及びサーバ装置については、装置の遠隔管理，電子商取引等の目的に応じて種々の構成をとることができる。例えば、遠隔管理の場合には、プリンタ，ＦＡＸ装置，コピー機，スキャナ，デジタル複合機等の画像処理装置を始め、ネットワーク家電，自動販売機，医療機器，電源装置，空調システム，ガス・水道・電気等の計量システム等の電子装置を被管理装置であるサーバ装置とし、これらの被管理装置から情報を収集したり、コマンドを送って動作させたりするための管理装置をクライアント装置とすることが考えられる。

【００５２】

しかし、クライアント装置及びサーバ装置は、少なくともそれぞれＣＰＵ，ＲＯＭ，ＲＡＭ，ネットワークを介して外部装置と通信するための通信Ｉ／Ｆ、および認証処理に必要な情報を記憶する記憶手段を備え、ＣＰＵがＲＯＭ等に記憶した所要の制御プログラムを実行することにより、装置をクライアントあるいはサーバとして機能させることができるものとする。

なお、この通信には、有線，無線を問わず、ネットワークを構築可能な各種通信回線（通信経路）を採用することができる。証明書管理装置１０との間の通信についても同様である。

【００５３】

図２には、上述のように、各装置のこの発明の特徴となる部分の機能構成を示している。

まず、証明書管理装置１０は、証明用鍵作成部２１，証明書発行部２２，証明書管理部２３，証明書更新部２４，通信機能部２５，構成記憶部２６，更新順制御部２７を備えている。

証明用鍵作成部２１は、デジタル署名の作成に用いる証明用私有鍵であるルート私有鍵と、そのデジタル証明書の正当性を確認するための、ルート私有鍵と対応する証明用公開鍵（証明鍵）であるルート鍵とを作成する証明用鍵作成手段の機能を有する。

【００５４】

証明書発行部２２は、サーバ装置３０とクライアント装置４０との間の認証処

理に用いる認証情報であるクライアント公開鍵およびサーバ公開鍵にデジタル署名を付して、デジタル証明書であるクライアント公開鍵証明書およびサーバ公開鍵証明書として発行する証明書発行手段の機能を有する。また、クライアント公開鍵，クライアント私有鍵，サーバ公開鍵，サーバ私有鍵の作成及び、ルート鍵にデジタル署名を付したデジタル証明書であるルート鍵証明書の作成も、この証明書発行部２２の機能である。

証明書管理部２３は、証明書発行部２２が発行したデジタル証明書、その作成に用いたルート私有鍵、およびそのルート私有鍵と対応するルート鍵を管理する証明書管理手段の機能を有する。そして、これらの証明書や鍵を、その有効期限や発行先、ＩＤ、更新の有無等の情報と共に記憶する。

【００５５】

証明書更新部２４は、ルート鍵の更新を行う場合に、有効なルート私有鍵の各々について、新たなルート私有鍵（新ルート私有鍵）及びこれと対応する新たなルート鍵（新ルート鍵）を証明用鍵作成部２１に作成させ、これらを更新する証明用鍵更新手段の機能を有する。さらに、この更新に当たって、証明書発行部２２に新ルート私有鍵を用いてデジタル署名を付した新たなクライアント公開鍵証明書（新クライアント公開鍵証明書）、新たなサーバ公開鍵証明書（新サーバ公開鍵証明書）及び新たなルート鍵証明書（新ルート鍵証明書）を発行させ、通信機能部２５によってこれらをサーバ装置３０及びクライアント装置４０に送信させ、サーバ装置３０及びクライアント装置４０にこれらの更新を要求させる機能も有する。また、詳細は後述するが、更新に必要な各処理の手順や進捗状況の管理は、更新順制御部２７が行う。

【００５６】

通信機能部２５は、ネットワークを介して外部装置と通信する機能を有し、証明書管理部２３の指示に応じて必要なデータをサーバ装置３０及びクライアント装置４０に送信したり、受信したデータを証明書更新部２４に渡したりする。

構成記憶部２６は、証明書管理装置１０がデジタル証明書の管理を行う対象であるクライアント・サーバシステムを構成する各ノード（ここではサーバ装置３０及びクライアント装置４０）について、少なくとも該ノードの通信相手及び該

通信相手との間でクライアントとサーバのいずれとして機能するかの情報を記憶し、構成記憶手段の機能を有する。ここではさらに、各ノードが相互認証に使用する私有鍵、公開鍵証明書、およびルート鍵証明書のＩＤ及び、鍵や証明書の更新状態に関する情報も記憶するものとする。

更新順制御部２７は、ルート鍵の更新の必要が生じた場合に、構成記憶部２６に記憶している情報をもとに、証明書更新部２４による鍵や証明書の更新手順を定め、証明書更新部２４に更新動作を行わせると共にこれを制御する更新順制御手段として機能する。

そして、これらの各部の機能は、図１に示したＣＰＵ１１が所要の制御プログラムを実行して証明書管理装置１０の各部の動作を制御することにより実現される。

【００５７】

一方、サーバ装置３０は、証明書記憶部３１，通信機能部３２，サーバ機能部３３を備えている。

証明書記憶部３１は、ＳＳＬによる相互認証に用いる鍵を記憶する機能を有し、図４６に示したルート鍵証明書、サーバ私有鍵、およびサーバ公開鍵証明書を記憶する。

通信機能部３２は、ネットワークを介して外部装置と通信する機能を有し、受信したデータをサーバ機能部３３に渡し、またサーバ機能部３３の指示に従ってデータを外部装置に送信する。

【００５８】

サーバ機能部３３は、クライアント装置４０から受信した要求に対して所要の処理を行って応答を返すサーバとしての機能を有する。また、以下に詳述するが、証明書管理装置１０から受信した証明書更新等の要求に対しても、所要の処理を行って応答を返す。

そして、これらの各部の機能は、サーバ装置３０のＣＰＵが所要の制御プログラムを実行してサーバ装置３０の各部の動作を制御することにより実現される。

【００５９】

また、クライアント装置４０は、証明書記憶部４１，通信機能部４２，クライ

アント機能部４３を備えている。

証明書記憶部４１は、ＳＳＬによる相互認証に用いる鍵を記憶する機能を有し、図４６に示したルート鍵証明書、クライアント私有鍵、およびクライアント公開鍵証明書を記憶する。

通信機能部４２は、ネットワークを介して外部装置と通信する機能を有し、受信したデータをクライアント機能部４３に渡し、またクライアント機能部４３の指示に従ってデータを外部装置に送信する。

【００６０】

クライアント機能部４３は、ユーザからの操作、図示しないセンサが検出した状態変化、あるいは図示しないタイマによって計測した所定時間経過等をトリガとして、サーバ装置３０に対して所要の要求を送信し、サーバ装置３０からこれに対する応答を受信した場合にはその内容に従った処理を行うクライアントとしての機能を有する。また、以下に詳述するが、応答として証明書管理装置１０からの証明書更新等の要求を受信した場合には、所要の処理を行って応答を返す。

そして、これらの各部の機能は、クライアント装置４０のＣＰＵが所要の制御プログラムを実行してクライアント装置４０の各部の動作を制御することにより実現される。

【００６１】

なお、このデジタル証明書管理装置において、証明書管理装置１０が直接通信可能なのは、クライアント・サーバシステムを構成する装置のうちサーバ装置３０のみであり、証明書管理装置１０からクライアント装置４０に対する要求は、サーバ装置３０が中継して送るものとする。クライアント装置４０から証明書管理装置１０への応答も、同様である。

また、上記のサーバ装置３０及びクライアント装置４０には、工場出荷時あるいはそれに順ずる時期、少なくともユーザが相互認証処理の運用を開始する前に、初めのルート鍵を記憶させておくものとする。このとき、公開鍵証明書及び秘密鍵も共に記憶させるようにするとよい。

【００６２】

次に、このような基本的な機能を有する図２に示したデジタル証明書管理シス

テムにおけるこの発明の特徴に関連する処理である、ルート鍵更新処理およびそのために必要な構成について説明する。

なお、以下の説明に用いるシーケンス図に記載するサーバ装置３０とクライアント装置４０と間の通信処理に際しては、個々に図示はしていないが、通信の確立前に従来の技術の項で図４６を用いて説明したＳＳＬによる相互認証を行い、認証が成功した場合のみデータの転送を行うものとする。そして、この相互認証処理に支障を来さないようにルート鍵証明書を更新可能であることが、この発明の特徴である。以下の実施形態についても同様である。

またここでは、証明書管理装置１０とサーバ装置３０との間の通信は、直通回線等の、安全（データの改竄や盗聴がなされないこと）を確保できる通信経路を介して行うものとする。

【００６３】

また、ここで説明するルート鍵更新処理は、この発明のデジタル証明書管理方法の第１の実施形態に係る処理であり、図４乃至図１０のシーケンス図に示す処理を、図１１のフローチャートに示す順番で実行するものである。そこで、まず図４乃至図１０の各シーケンス図に示す処理の内容を説明してから、図１１を用いてその実行順について説明する。以下の各図に示す処理は、証明書管理装置１０，サーバ装置３０，クライアント装置４０の各ＣＰＵが、所要の制御プログラムを実行することによって行うものである。

【００６４】

まず図４のシーケンス図に処理Ｓとしてルート鍵証明書作成処理を示す。

この処理においては、証明書管理装置１０は、ステップＳ１０１で、有効なルート私有鍵について、新たなルート私有鍵とルート鍵のペアを作成する。ここで、「有効な」ルート私有鍵とは、その時点でクライアント・サーバシステムにおける相互認証に使用中のルート私有鍵という意味であり、より正確には、そのルート私有鍵を用いてデジタル署名を付した証明書が、認証処理に用いられる状態でサーバ装置３０又はクライアント装置４０に記憶されているものをいうものとする。過去に作成した私有鍵が有効か否かは、証明書管理部２３に記憶している公開鍵証明書及びルート鍵証明書の有効期限やその更新の有無の情報や、構成記

憶部２６に記憶している各ノードが使用している公開鍵証明書及びルート鍵証明書のＩＤの情報、および証明書に含まれる、デジタル署名に使用したルート私有鍵の識別情報等の情報を基に判断することができる。また、新たな鍵と置き換えられるべきそれまでの鍵を、「従前の」鍵と呼ぶことにする。証明書についても同様である。

そして、ステップＳ１０２で、ステップＳ１０１で作成した新ルート鍵に従前のルート私有鍵を用いたデジタル署名を付し、第１の証明鍵証明書である配布用ルート鍵証明書を作成する。

以上がルート鍵証明書作成処理である。

【００６５】

次に、図５のシーケンス図に処理１としてサーバ装置のルート鍵証明書記憶処理を示す。

この処理においては、まずステップＳ１１１で、証明書管理装置１０がサーバ装置３０に対して、図４のステップＳ１０２で作成した配布用ルート鍵証明書と共に、その更新要求を送信する。この処理において、証明書管理装置１０のＣＰＵ１１が第２の更新要求手段として機能する。

【００６６】

サーバ装置３０は、この要求を受け取ると、ステップＳ１１２で従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認する。上述のように、配布用ルート鍵証明書には、従前のルート私有鍵を用いたデジタル署名を付しているので、従前のルート鍵を用いてその内容を復号化し、確かに証明書管理装置１０によって発行されたものであることを確認できる。また、このとき、従来の技術の項で図４７を用いて説明したようにルート鍵が損傷や改竄等を受けていないことも確認できる。従って、このような配布用ルート鍵証明書を用いることにより、受け取ったルート鍵の正当性を人手によらず確認できることになる。

そして、これが確認できると、次のステップＳ１１３で配布用ルート鍵証明書を証明書記憶部３１に記憶する。このとき、まだ従前のルート鍵証明書は消去しない。従って、証明書記憶部３１には２つのルート鍵証明書が記憶された状態となる。

【００６７】

この状態で認証処理を行う場合、受信した公開鍵証明書の正当性を確認する際には、２つのルート鍵証明書を順次用いて確認を試み、どちらかのルート鍵証明書を用いて確認が成功すれば、正当性が確認できたものとする。従って、新旧どちらのルート私有鍵を用いてデジタル署名を付したデジタル証明書であっても、その正当性を確認することができる。なお、配布用ルート鍵証明書を認証処理に使用する際の、ルート鍵に破損や改竄がないことの確認は、従前のルート鍵証明書を用いて行うことができる。これらのステップＳ１１２及びＳ１１３において、サーバ装置３０のＣＰＵが第２のサーバ側更新手段として機能する。

サーバ装置３０はその後、ステップＳ１１４で証明書管理装置１０に対して更新要求に対する応答として結果通知を返し、配布用ルート鍵証明書の記憶が成功していればその旨を、何らかの理由で失敗していればその旨を伝える。

以上がサーバ装置のルート鍵証明書記憶処理である。

【００６８】

次に、図６のシーケンス図に処理２としてクライアント装置のルート鍵証明書記憶処理を示す。

この処理においては、まずステップＳ１２１で、証明書管理装置１０がサーバ装置３０に対して、図４のステップＳ１０２で作成した配布用ルート鍵証明書と共に、その更新要求をクライアント装置４０に送信するよう要求する更新要求送信要求を送信する。サーバ装置３０は、これに応じてクライアント装置４０に対して配布用ルート鍵証明書とその更新要求とを送信するのであるが、サーバ装置３０側から送信要求を行うことはできない。そこで、クライアント装置４０が所定のタイミングで定期的にサーバ装置３０に対してポーリングして通信を要求するようにし（Ｓ１２２）、これに対する応答として配布用ルート鍵証明書とその更新要求とを送信するようにしている（Ｓ１２３）。

【００６９】

なお、クライアント装置４０がサーバ装置３０に対するポーリングや接続要求をＨＴＴＰリクエストとして送信し、サーバ装置３０からクライアント装置４０に対して送信する要求やデータをこれに対する応答であるＨＴＴＰレスポンスと

して送信するようにするとよい。このようにすれば、クライアント装置４０がファイアウォールの内側に設置されている場合でも、これを越えてサーバ装置３０からクライアント装置４０にデータを転送することができる。

【００７０】

ファイアウォールを越える手段はこれに限られるものではなく、例えば、ＳＭＴＰ（Simple Mail Transfer Protocol）を利用して、送信したいデータを記載あるいは添付したメールを送信することも考えられる。ただし、信頼性の面ではＨＴＴＰが優れている。

以上の処理により、証明書管理装置１０からクライアント装置４０に、サーバ装置３０を介して配布用ルート鍵証明書とその更新要求とが送信されることになり、ステップＳ１２１の処理においては、証明書管理装置１０のＣＰＵ１１が第１の更新要求手段として機能する。

【００７１】

クライアント装置４０は、この要求を受け取ると、ステップＳ１２４で従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認する。そして、これが確認できると、次のステップＳ１２５で配布用ルート鍵証明書を証明書記憶部４１に記憶する。このとき、まだ従前のルート鍵証明書は消去しない。これらの確認と記憶の詳細については、図５のステップＳ１１２及びＳ１１３の場合と同様であり、これらのステップにおいて、クライアント装置４０のＣＰＵが第２のクライアント側更新手段として機能する。

クライアント装置４０はその後、ステップＳ１２６で証明書管理装置１０に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置３０に対して送信し、サーバ装置３０がステップＳ１２７で証明書管理装置に対して送信する。

以上がクライアント装置のルート鍵証明書記憶処理である。

【００７２】

次に、図７のシーケンス図に処理３としてクライアント装置の公開鍵証明書記憶処理を示す。

この処理においてはまずステップＳ１３１で、証明書管理装置１０が、クライ

アント装置４０に対して発行してあるクライアント公開鍵に、新ルート私有鍵を用いたデジタル署名を付して新クライアント公開鍵証明書を作成する。なお、クライアント私有鍵は更新しないので、クライアント公開鍵自体も更新する必要はない。

【００７３】

そしてステップＳ１３２で、証明書管理装置１０がサーバ装置３０に対して、ステップＳ１３１で作成した新クライアント公開鍵証明書と共に、その更新要求をクライアント装置４０に送信するよう要求する更新要求送信要求を送信する。サーバ装置３０は、これに応じて、図６のステップＳ１２２及びＳ１２３の場合と同様に、クライアント装置４０からのポーリング（Ｓ１３３）に対する応答として新クライアント公開鍵証明書とその更新要求とを送信するようにしている（Ｓ１３４）。

以上の処理により、証明書管理装置１０からクライアント装置４０にサーバ装置３０を介して新クライアント公開鍵証明書とその更新要求とが送信されることになり、ステップＳ１３２の処理においては、証明書管理装置１０のＣＰＵ１１が第１の更新要求手段として機能する。

【００７４】

クライアント装置４０は、この要求を受け取るとステップＳ１３５で、図６のステップＳ１２５で記憶した配布用ルート鍵証明書を用いて新クライアント公開鍵証明書の正当性を確認する。上述のように、新クライアント公開鍵証明書には、新ルート私有鍵を用いたデジタル署名を付しているので、配布用ルート鍵証明書に含まれる新ルート鍵を用いてその内容を復号化し、確かに証明書管理装置１０によってクライアント装置４０に対して発行されたものであることを確認できる。そして、これが確認できると、次のステップＳ１３６で新クライアント公開鍵証明書を証明書記憶部４１に記憶する。これらのステップＳ１３５及びＳ１３６において、クライアント装置４０のＣＰＵが第１のクライアント側更新手段として機能する。

【００７５】

このとき、まだ従前のクライアント公開鍵証明書は消去しない。従って、証明

書記憶部４１には２つのクライアント公開鍵証明書が記憶された状態となる。この状態で認証処理を行い、通信相手に対して公開鍵証明書を送信する場合には、まず新公開鍵証明書を送信するものとする。

この場合、通信相手が既に新ルート鍵を（配布用ルート鍵証明書又は後述する新ルート鍵証明書として）記憶していれば、新公開鍵証明書のデジタル署名を復号化できるので、問題なく認証を受けることができる。一方、通信相手がまた新ルート鍵を記憶していない場合には、新公開鍵証明書のデジタル署名を復号化できず、認証が失敗した旨の応答を受けることになる。しかしこの場合でも、再度通信を要求し、この際に従前の公開鍵証明書を送信するようにすれば、従前のルート鍵によってそこに付されたデジタル署名を復号化できるので、問題なく認証を受けることができる。

【００７６】

従って、２つの公開鍵証明書を記憶しておけば、通信相手が新ルート鍵を記憶していない場合に多少のオーバーヘッドが生じることはあるが、問題なく相互認証を行うことができる。なお、２つの公開鍵証明書に含まれる公開鍵本体は同じものであるので、クライアント私有鍵を用いて暗号化したデータの復号化は、どちらの公開鍵証明書を用いた場合でも同じように行うことができる。

クライアント装置４０はその後、ステップＳ１３７で証明書管理装置１０に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置３０に対して送信し、サーバ装置３０がステップＳ１３８で証明書管理装置に対して送信する。

以上がクライアント装置の公開鍵証明書記憶処理である。

【００７７】

次に、図８のシーケンス図に処理４としてサーバ装置の公開鍵証明書記憶処理を示す。

この処理においてはまずステップＳ１４１で、証明書管理装置１０が、クライアント装置４０に対して発行してあるサーバ公開鍵に、新ルート私有鍵を用いたデジタル署名を付して新サーバ公開鍵証明書を作成する。サーバ公開鍵自体の更新が不要であることは、上述のクライアント公開鍵の場合と同様である。

そしてステップＳ１４２で、証明書管理装置１０がサーバ装置３０に対して、新サーバ公開鍵証明書と共にその更新要求を送信する。この処理において、証明書管理装置１０のＣＰＵ１１が第２の更新要求手段として機能する。

【００７８】

サーバ装置３０は、この要求を受け取るとステップＳ１４３で、図５のステップＳ１１３で記憶した配布用ルート鍵証明書を用いて新公開鍵証明書の正当性を確認する。この点については、図７のステップＳ１３５の場合と同様である。そして、これが確認できると、次のステップＳ１４４で新サーバ公開鍵証明書を証明書記憶部４１に記憶し、従前のサーバ公開鍵証明書と置き換える。これらのステップＳ１４３及びＳ１４４において、サーバ装置３０のＣＰＵが第１のサーバ側更新手段として機能する。

【００７９】

ところで、サーバ装置３０の場合には、クライアント装置４０の場合と異なり、新公開鍵証明書を記憶させる場合に従前のものに追加するのではなくこれと置き換える必要があるのであるが、ここでこの点について説明する。

サーバ装置３０の場合には、クライアント装置４０から接続要求があった場合に公開鍵証明書をクライアント装置４０に送信するのであるが、サーバ公開鍵証明書を複数記憶していたとすると、送信毎にそのうちいずれかを選択して送信することになる。そして、クライアント装置４０側でデジタル証明書を復号化できないようなサーバ公開鍵証明書を送信してしまった場合には、認証は失敗することになる。例えば、クライアント装置４０が新ルート鍵を記憶する前に新サーバ公開鍵証明書を送信した場合等である。

【００８０】

たとえ失敗したとしても、次に接続要求があった場合にもう一方のサーバ公開鍵証明書を送信すればよいという考え方もあるが、不特定多数のクライアント装置から任意のタイミングで接続要求を受け得るサーバ装置の場合、クライアント装置毎に送信すべきサーバ公開鍵証明書を選択することは、現実的ではない。また、クライアント装置がどのような装置であるかは、サーバ装置側では認証が済むまで通常わからないので、最初に送信するサーバ公開鍵証明書を適切に選択す

ることも困難である。従って、サーバ装置にはサーバ公開鍵証明書を１つだけ記憶させ、クライアント装置から接続要求を受けた場合には常にこれを送信するようにする必要があるのである。

【００８１】

従って、サーバ装置３０では新サーバ公開鍵証明書を記憶させた時点で従前のサーバ公開鍵証明書は削除してしまうので、クライアント装置４０に新ルート鍵を記憶させる前にこれを行ってしまうと、クライアント装置側でサーバ公開鍵証明書のデジタル署名を復号化できなくなり、相互認証を行えなくなってしまう。そこで、サーバ装置３０の公開鍵証明書記憶処理は、クライアント装置のルート鍵証明書記憶処理の完了後に行う必要がある。

以上のようなステップＳ１４４の終了後、サーバ装置３０はステップＳ１４５で証明書管理装置１０に対して更新要求に対する応答として結果通知を返し、新サーバ公開鍵証明書の記憶が成功していればその旨を、何らかの理由で失敗していればその旨を伝える。

以上がサーバ装置の公開鍵証明書記憶処理である。

【００８２】

次に、図９のシーケンス図に処理５としてサーバ装置のルート鍵証明書書き換え処理を示す。

この処理においてはまずステップＳ１５１で、証明書管理装置１０が、新ルート鍵に新ルート私有鍵を用いたデジタル署名を付して第２の証明鍵証明書として新ルート鍵証明書を作成する。そして、ステップＳ１５２で証明書管理装置１０がサーバ装置３０に対して、新ルート鍵証明書と共にその更新要求を送信する。この処理においても、証明書管理装置１０のＣＰＵ１１が第２の更新要求手段として機能する。

【００８３】

サーバ装置３０は、この要求を受け取ると、ステップＳ１５３で配布用ルート鍵証明書を用いて新ルート鍵証明書の正当性を確認する。上述のように、新ルート鍵証明書には、新ルート私有鍵を用いたデジタル署名を付しているので、配布用ルート鍵証明書に含まれる新ルート鍵を用いてその内容を復号化し、確かに証

明書管理装置１０によって発行されたものであることを確認できる。

そして、これが確認できると、次のステップＳ１５４で新ルート鍵証明書を証明書記憶部３１に記憶する。そして、配布用ルート鍵証明書及び従前のルート鍵証明書を削除して廃棄し、ルート鍵証明書を新たなものに書き換えてしまう。このようにすると、従前のルート私有鍵を用いてデジタル署名を付したデジタル証明書は復号化できなくなってしまうが、クライアント装置４０に新クライアント公開鍵証明書を記憶させた後であれば、クライアント装置４０から送られてくる公開鍵証明書の確認には支障がないので、認証処理に支障を来すことはない。

【００８４】

サーバ装置３０はその後、ステップＳ１５５で証明書管理装置１０に対して更新要求に対する応答として結果通知を返し、新ルート鍵証明書の記憶が成功していればその旨を、何らかの理由で失敗していればその旨を伝える。

以上がサーバ装置のルート鍵証明書書き換え処理である。

【００８５】

次に、図１０のシーケンス図に処理６としてクライアント装置のルート鍵証明書書き換え処理を示す。

この処理においてはまずステップＳ１６１で、証明書管理装置１０が、新ルート鍵に新ルート私有鍵を用いたデジタル署名を付して第２の証明鍵証明書として新ルート鍵証明書を作成する。これは、図９のステップＳ１５１で作成するものと同じであるので、ここで作成したものを流用してもよい。逆に図９のステップＳ１５１で、このステップＳ１６１で作成したものを流用してもよい。

【００８６】

そしてステップＳ１６２で、証明書管理装置１０がサーバ装置３０に対して、ステップＳ１６１で作成した新ルート鍵証明書と共に、その更新要求をクライアント装置４０に送信するよう要求する更新要求送信要求を送信する。サーバ装置３０は、これに応じて、図６のステップＳ１２２及びＳ１２３の場合と同様に、クライアント装置４０からのポーリング（Ｓ１６３）に対する応答として新ルート鍵証明書とその更新要求とを送信するようにしている（Ｓ１６４）。

以上の処理により、証明書管理装置１０からクライアント装置４０にサーバ装

置３０を介して新ルート鍵証明書とその更新要求とが送信されることになり、ステップＳ１６２の処理においても、証明書管理装置１０のＣＰＵ１１が第１の更新要求手段として機能する。

【００８７】

クライアント装置４０は、この要求を受け取ると、ステップＳ１６５で配布用ルート鍵証明書を用いて新ルート鍵証明書の正当性を確認する。そして、これが確認できると、次のステップＳ１６６で新ルート鍵証明書を証明書記憶部４１に記憶する。そして、配布用ルート鍵証明書及び従前のルート鍵証明書を削除して廃棄し、ルート鍵証明書を新たなものに書き換えてしまう。これらの処理については、図９のステップＳ１５３及びＳ１５４の場合と同様である。ただし、クライアント装置４０への新クライアント公開鍵証明書の記憶が済んでいれば、ステップＳ１６６で従前のクライアント公開鍵証明書も同時に廃棄してしまってよい。

【００８８】

クライアント装置４０はその後、ステップＳ１６７で証明書管理装置１０に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置３０に対して送信し、サーバ装置３０がステップＳ１６８で証明書管理装置に対して送信する。

以上がクライアント装置のルート鍵証明書書き換え処理である。

【００８９】

以上の図４乃至図１０に示した各処理の実行タイミングは、証明書管理装置１０の更新順制御部２７が構成記憶部２６に記憶している情報をもとに更新手順を作成して管理する。そして、その更新手順はここでは図１１に示すフローチャートのようなものになる。すなわち、ルート鍵の更新事由を検出した場合に、図１１のフローチャートに示す処理を開始し、まず図４に示した処理Ｓを実行し、その後処理１乃至処理６を実行する。なお、ルート鍵の更新事由としては、所定の有効期限の到来、管理者の指示等が考えられる。管理者が更新の指示を行う場合としては、ルート私有鍵の第３者への漏洩が判明した場合等が考えられる。

また、図１１において、矢印の先の処理は、矢印の根元側の処理が全て完了し

てから開始する。破線で示した矢印については、その条件は必須ではないが考慮した方が好ましいということを示す。

【００９０】

具体的には、処理１及び処理２は処理Ｓの完了後に開始する。処理３は、処理２の完了後に開始するが、処理１も完了した後に開始する方が好ましい。処理４は、処理１及び処理２の完了後に開始する。処理５は、処理１及び処理３の完了後に開始する。処理６は、処理２及び処理４の完了後に開始する。そして、処理３乃至６が全て完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

【００９１】

なお。各処理は、更新要求に対する更新成功の応答を受け取った場合に完了したものとすることができる。更新失敗の応答を受け取った場合や処理がタイムアウトした場合には、再度同じ処理を試みるとよいが、所定回数続けて失敗した場合には更新処理全体が失敗したものとするとよい。ルート鍵更新処理を図１１に示す手順で行う場合、サーバ装置３０とクライアント装置４０とは処理のどの時点であっても互いにＳＳＬによる相互認証を行うことができるので、このように更新処理が途中で中断してしまっても、サーバ装置３０とクライアント装置４０との間の通信に大きな支障はない。従って、更新処理が失敗した場合に時間をかけて失敗の原因を特定した上で改めて更新処理を行っても、特に大きな問題はない。以後の各実施形態についても同様である。

【００９２】

このデジタル証明書管理システムにおいては、ルート鍵更新処理をこのような手順で行うことにより、サーバ装置３０とクライアント装置４０との間の相互認証処理に大きな影響を与えることなく、ルート鍵を自動制御で更新することができる。従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおけるＳＳＬによる相互認証を、低コストで運用することができる。

証明書管理装置１０とサーバ装置３０との間には、これとは別の安全な通信経路を設ける必要があるが、証明書管理装置１０と１つの装置のみとの間に設けれ

ばよいので、特に大きな負担にはならない。証明書管理装置１０とサーバ装置３０とが物理的に近接している場合には専用ケーブルで結ぶ等してこのような経路を設けることは容易であり、この実施形態はこのような場合に好ましいものであると言える。

【００９３】

図１１に示す処理手順において、この発明の特徴となるのは、まず、処理４（サーバ装置の公開鍵証明書記憶処理）を処理２（クライアント装置のルート鍵証明書記憶処理）の後で、すなわちクライアント装置４０から配布用ルート鍵証明書を記憶した旨の応答があった後で実行する点である。

処理４の説明において上述したように、サーバ装置３０については公開鍵証明書を同時に２つ記憶させると不都合が生じるので、新サーバ公開鍵証明書を記憶させる際には従前のものを廃棄する必要があるのであるが、このような書き換えを行ってしまっても、クライアント装置４０に新ルート鍵を記憶させた後であれば、認証処理に支障が生じることがない。

【００９４】

また、処理３（クライアント装置の公開鍵証明書記憶処理）を処理１（サーバ装置のルート鍵証明書記憶処理）の後で、すなわちサーバ装置３０から配布用ルート鍵証明書を記憶した旨の応答があった後で実行するようにするとよい。

処理３の説明で上述したように、クライアント装置４０に新クライアント公開鍵証明書を記憶させた時点でサーバ装置３０に新ルート鍵が記憶されていないと、サーバ装置３０に新ルート鍵が記憶されるまで通信にオーバーヘッドが生じ、効率が悪くなってしまうためである。

【００９５】

処理５と処理６については、これらは必須の処理ではないが、従前のルート鍵証明書や公開鍵証明書をいつまでも記憶させておくとすると、記憶容量を無駄に消費することになる。鍵や証明書の記憶には、信頼性の高い記憶手段を用いることが好ましく、従って容量当たりのコストが高いので、この点は大きな問題になる。また、配布用ルート鍵証明書は、自己署名形式でないので、使用する際に従前のルート鍵証明書を参照する必要があり、処理効率が悪い。そこで、処理５と

処理６を行って、ルート鍵証明書を自己署名形式のものにすると共に、従前の証明書を廃棄するようにするとよい。

【００９６】

ルート鍵証明書を自己署名形式のものに書き換えるだけであれば、配布用ルート鍵証明書を記憶させた直後に、例えば処理５の場合には処理１の完了直後に行ってもよいのであるが、この時点では必ずしも従前のルート鍵証明書を廃棄できない。そして、この削除タイミングはサーバ装置３０側では決定することができないので、処理３の終了後に再度従前のルート鍵証明書を廃棄する要求を行う必要が生じてしまう。従って、処理１と処理３の完了後に処理５を行うことが、処理の簡略化の点から好ましい。処理６についても、同様の理由から処理２と処理４の完了後に行うことが好ましい。

【００９７】

なお、ルート鍵は一旦記憶してしまえば一般に外部に送信する必要はないので、その後の破損や改竄は考えにくいことから、ルート鍵証明書ではなく、鍵部分のみを記憶することも考えられる。このような場合には、配布用ルート鍵証明書に含まれる新ルート鍵を記憶してしまえばよいので、証明書管理装置１０から新ルート鍵証明書を別途送信する必要はない。そこで、このような場合、処理５，処理６においては、新ルート鍵証明書を送信せず、従前のルート鍵の廃棄のみを要求するようにすればよい。また、ルート鍵を使用する際に、デジタル署名の確認を行わないようにする場合についても同様である。

なお、この実施形態において、サーバ装置３０からクライアント装置４０への送信は、クライアント装置４０からのポーリングに対する応答として行う例について説明したが、サーバ装置３０がクライアントとしても機能できるようにし、クライアント装置４０がサーバとしても機能できるようにし、これらの機能によって、サーバ装置３０からクライアント装置４０へデータや要求を直接送信できるようにしてもよい。このような場合は、クライアント装置４０によるポーリングは不要である。この点は、以下の実施形態においても同様である。

【００９８】

〔第２の実施形態：図１２乃至図１９〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第２の実施形態の構成について説明する。この実施形態においても、各１つのクライアント及びサーバによってクライアント・サーバシステムを構成しており、この実施形態は、この発明を最も基本的なシステムに適用した第１の実施形態とは別の例である。

このデジタル証明書管理システムを構成する各装置の、この発明の特徴となる部分の機能構成を、図２と対応する図１２の機能ブロック図に示す。この図において、図２と対応する部分には同一の符号を付している。

【００９９】

この図からわかるように、このデジタル証明書管理システムにおいてはまず、証明書管理装置１０をクライアント・サーバシステムを構成する装置のうちクライアント装置４０のみと直接通信可能とし、証明書管理装置１０からサーバ装置３０に対する要求は、クライアント装置４０が中継して送るものとした点が第１の実施形態と異なる。

また、クライアント装置４０にもサーバ機能部４４を設けた点も、第１の実施形態の場合と異なるが、このサーバ機能部４４は、受信した要求に対して所要の処理を行って応答を返すサーバとしての機能を有し、証明書管理装置１０との通信のために設けたものである。クライアント装置４０がクライアント機能部４３しか有しないとすると、証明書管理装置１０からクライアント装置４０にデータや要求等を送信する場合に、クライアント装置４０からのポーリングを待つ必要が生じてしまう。

【０１００】

しかし、ルート鍵の更新処理は頻繁に行われるものではなく、例えば年に１回程度であるので、このためにクライアント装置４０が証明書管理装置１０に対して定期的にポーリングを行うとすると、ほとんどの通信が無駄になることになる。そこで、クライアント装置４０にサーバ機能部４４を設け、証明書管理装置１０側から通信を要求できるようにしたものである。このサーバ機能部４４の機能も、クライアント装置４０のＣＰＵが所要の制御プログラムを実行してクライア

ント装置４０の各部の動作を制御することにより実現されるものである。

【０１０１】

ただし、クライアント・サーバシステムを構成するサーバ装置３０との関係においては、クライアント装置４０は常にクライアントとして機能する。従って、証明書管理装置１０からサーバ装置３０への通信を仲介する場合には、通信機能部４２が証明書管理装置１０から受信したデータや要求を、サーバ機能部４４が受け取り、これをクライアント機能部４３に渡して、クライアント機能部４３の指示に基づいてサーバ装置３０に対する通信を要求してサーバ装置３０に送信することになる。サーバ装置３０からの応答を証明書管理装置１０に返す場合には、この逆の処理となる。

【０１０２】

これらの変更に伴ってルート鍵更新処理のシーケンスは変更されるが、それ以外の点については第１の実施形態と同様であるので、説明を省略する。

なおここでも、証明書管理装置１０とクライアント装置４０との間の通信は、直通回線等の、安全を確保できる通信経路を介して行うものとする。ただし、この実施形態の場合には、証明書管理装置１０とクライアント装置４０との間の通信にＳＳＬを用いることも可能であるが、この場合の構成については変形例として後述する。

【０１０３】

このデジタル証明書管理システムにおけるルート鍵更新動作は、この発明のデジタル証明書管理方法の第２の実施形態に係る動作であり、図１３乃至図１８のシーケンス図に示す処理及び図４を用いて上述した処理Ｓを、図１９のフローチャートに示す順番で実行するものである。そこで、まず図１３乃至図１８の各シーケンス図に示す処理の内容を説明してから、図１８を用いてその実行順について説明する。以下の各図に示す処理は、証明書管理装置１０，サーバ装置３０，クライアント装置４０の各ＣＰＵが、所要の制御プログラムを実行することによって行うものである。

【０１０４】

まず、図１３のシーケンス図に処理１１としてサーバ装置のルート鍵証明書記

憶処理を示す。

この処理は、図５に示した処理１と同じ目的の処理であるが、ここでは証明書管理装置１０と直接通信する装置がクライアント装置４０であるため、手順が若干異なるものとなっている。

【０１０５】

すなわち、まずステップＳ２１１で、証明書管理装置１０がクライアント装置４０に対して、図４のステップＳ１０２で作成した配布用ルート鍵証明書と共に、その更新要求をサーバ装置３０に送信するよう要求する更新要求送信要求を送信する。そしてクライアント装置４０は、これに応じてサーバ装置３０に対して配布用ルート鍵証明書とその更新要求とを送信する（Ｓ２１２）。クライアント装置４０はサーバ装置３０に対して通信を要求できるので、図５の場合のようにポーリングを待つ必要はない。

以上の処理により、証明書管理装置１０からサーバ装置３０にクライアント装置４０を介して配布用ルート鍵証明書とその更新要求とが送信されることになり、ステップＳ２１１の処理においては、証明書管理装置１０のＣＰＵ１１が第２の更新要求手段として機能する。

【０１０６】

サーバ装置３０は、ステップＳ２１２で送信されてきた更新要求を受け取ると、ステップＳ２１３で従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認し、これが確認できると、次のステップＳ２１４で配布用ルート鍵証明書を証明書記憶部３１に記憶する。これらの処理は、図５のステップＳ１１２及びＳ１１３の処理と全く同じである。

【０１０７】

サーバ装置３０はその後、ステップＳ２１５で証明書管理装置１０に対して更新要求に対する応答として結果通知を返すが、これはまずクライアント装置４０に対して送信し、クライアント装置４０がステップＳ２１６で証明書管理装置に対して送信する。なお、この結果通知は、クライアント装置４０から受信した更新要求に対する応答として送信することができるので、クライアント装置４０からのポーリングを待つ必要はない。

以上がこの実施形態におけるサーバ装置のルート鍵証明書記憶処理である。

【０１０８】

次に、図１４のシーケンス図に処理１２としてクライアント装置のルート鍵証明書記憶処理を示す。

この処理は、図６に示した処理２と同じ目的の処理であるが、処理１１の場合と同様に手順が若干異なるものとなっている。

この処理においては、まずステップＳ２２１で、証明書管理装置１０がクライアント装置４０に対して、図４のステップＳ１０２で作成した配布用ルート鍵証明書とその更新要求を送信する。この処理において、証明書管理装置１０のＣＰＵ１１が第１の更新要求手段として機能する。

【０１０９】

クライアント装置４０は、この要求を受け取ると、ステップＳ１２４で従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認し、これが確認できると、次のステップＳ１２５で配布用ルート鍵証明書を証明書記憶部４１に記憶する。これらの処理は、図６のステップＳ１２４及びＳ１２５の処理と全く同じである。

クライアント装置４０はその後、ステップＳ２２４で証明書管理装置１０に対して更新要求に対する応答として結果通知を返す。

以上がこの実施形態におけるクライアント装置のルート鍵証明書記憶処理である。

【０１１０】

以下、図１５に処理１３としてクライアント装置の公開鍵証明書記憶処理を、図１６に処理１４としてサーバ装置の公開鍵証明書記憶処理を、図１７に処理１５としてサーバ装置のルート鍵証明書書き換え処理を、図１８に処理１６としてクライアント装置のルート鍵証明書書き換え処理をそれぞれ示すが、これらは、第１の実施形態で図７乃至図１０を用いてそれぞれ説明した処理３乃至処理６と同じ目的の処理であり、証明書管理装置１０と直接通信する装置がクライアント装置４０であることに伴って、処理１１及び処理１２の場合と同様に通信手順を若干変更したのみである。そこで、これらの処理についての説明は省略する。

【０１１１】

また、以上の図１３乃至図１８に示した各処理及び図４に示した処理Ｓの実行タイミングは、証明書管理装置１０の更新順制御部２７が構成記憶部２６に記憶している情報をもとに更新手順を作成して管理する。そして、その更新手順はここでは図１９に示すフローチャートのようなものになる。すなわち、ルート鍵の更新を行う場合には、まず図４に示した処理Ｓを実行し、その後処理１１乃至処理１６を実行する。

図１９の記載から明らかなように、この第２の実施形態におけるルート鍵更新処理は、図１１に示した第１の実施形態の場合と対応する処理を、同様な順序で行うものである。そして、このことによる効果も、第１の実施形態の場合と同様である。

【０１１２】

すなわち、この第２の実施形態のデジタル証明書管理システムにおいては、ルート鍵更新処理をこのような手順で行うことにより、証明書管理装置１０がクライアント・サーバシステムを構成する装置のうちクライアント装置４０のみと通信可能な場合でも、第１の実施形態の場合と同様に、サーバ装置３０とクライアント装置４０との間の相互認証処理に大きな影響を与えることなくルート鍵を自動制御で更新することができる。従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおけるＳＳＬによる相互認証を、低コストで運用することができる。

また、この実施形態においては、クライアント装置４０にサーバ機能部４４を設ける必要はあるが、ルート鍵更新処理の手順にポーリング待ちを必要とする箇所がないため、処理を速やかに進め、短期間で完了させることができる。

【０１１３】

〔第３の実施形態：図２０乃至図２３〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第３の実施形態の構成について説明する。

このデジタル証明書管理システムは、ルート鍵更新処理の内容が第１の実施形態のデジタル証明書管理システムと異なるのみであり、装置の構成は第１の実施形態のものと同様であるのでその説明は省略する。

【０１１４】

このデジタル証明書管理システムにおけるルート鍵更新動作は、この発明のデジタル証明書管理方法の第３の実施形態に係る動作であり、図２０乃至図２３のシーケンス図に示す処理を、この順で実行するものである。以下の各図に示す処理は、証明書管理装置１０，サーバ装置３０，クライアント装置４０の各ＣＰＵが、所要の制御プログラムを実行することによって行うものである。

このデジタル証明書管理システムの証明書管理装置１０は、ルート鍵の更新事由を検出すると、図２０のシーケンス図に示す処理を開始する。

【０１１５】

図２０に示す処理は、第１の実施形態の説明において図４に示した処理Ｓと対応する処理Ｔである。そして、まずステップＳ３０１及びＳ３０２において、図４のステップＳ１０１及びＳ１０２の場合と同様に、有効なルート私有鍵について、新たなルート私有鍵とルート鍵のペアを作成すると共に、その新ルート鍵に従前のルート私有鍵を用いたデジタル署名を付し、第１の証明鍵証明書である配布用ルート鍵証明書を作成する。

そしてさらに、ステップＳ３０３において、図９のステップＳ１５１の場合と同様に、新ルート鍵に新ルート私有鍵を用いたデジタル署名を付して第２の証明鍵証明書として新ルート鍵証明書を作成する。

【０１１６】

その後、続いて図２１のシーケンス図に示す処理２１を行う。この処理は、第１の実施形態の説明において図６に示した処理２及び図７に示した処理３を併せ、さらに図１０に示した処理６の一部を加えた処理に相当する。

ここではまず、ステップＳ３１１で、図７のステップＳ１３１の場合と同様に、証明書管理装置１０がクライアント公開鍵に新ルート私有鍵を用いたデジタル署名を付して新クライアント公開鍵証明書を作成する。

【０１１７】

そしてステップＳ３１２で、証明書管理装置１０がサーバ装置３０に対して、図２０のステップＳ３０２で作成した配布用ルート鍵証明書と、図２０のステップＳ３０３で作成した新ルート鍵証明書と、ステップＳ３１１で作成した新クライアント公開鍵証明書と共に、これらについての更新要求をクライアント装置４０に送信するよう要求する更新要求送信要求を送信する。サーバ装置３０はこれに応じて、図６のステップＳ１２２及びＳ１２３の場合と同様に、クライアント装置４０からのポーリング（Ｓ３１３）に対する応答としてこれらの証明書とそれらについての更新要求とを送信するようにしている（Ｓ３１４）。

以上の処理により、証明書管理装置１０からクライアント装置４０にサーバ装置３０を介して上記の各証明書とそれらについての更新要求とが送信されることになり、ステップＳ３１２の処理においては、証明書管理装置１０のＣＰＵ１１が第１の更新要求手段として機能する。

【０１１８】

クライアント装置４０は、この要求を受け取ると、ステップＳ３１５及びＳ３１６で、図６のステップＳ１２４及びＳ１２５の場合と同様に、従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認し、これが確認できると配布用ルート鍵証明書を証明書記憶部４１に記憶する。このとき、まだ従前のルート鍵証明書は消去しない。

そしてさらにステップＳ３１７で、図１０のステップＳ１６５の場合と同様に、記憶した配布用ルート鍵証明書を用いて新ルート鍵証明書の正当性を確認する。そして、これが確認できると、次のステップＳ３１８で新ルート鍵証明書を証明書記憶部４１に記憶する。この時点で配布用ルート鍵は消去してしまってもよいが、ここでは記憶したままとする。

これらのステップＳ３１５乃至Ｓ３１８の処理において、クライアント装置４０のＣＰＵが第２のクライアント側更新手段として機能する。

【０１１９】

次に、ステップＳ３１９及びＳ３２０で、図７のステップＳ１３５及びＳ１３６の場合と同様に、新クライアント公開鍵証明書の正当性を確認し、これが確認できると、新クライアント公開鍵証明書を証明書記憶部４１に記憶する。ただし

、ここでは既に新ルート鍵証明書を記憶しているので、新クライアント公開鍵証明書の正当性は、配布用ルート鍵証明書ではなく新ルート鍵証明書を用いて行うことができる。これらのステップＳ３１９及びＳ３２０において、クライアント装置４０のＣＰＵが第１のクライアント側更新手段として機能する。

【０１２０】

このとき、まだ従前のクライアント公開鍵証明書は消去しない。従って、証明書記憶部４１には２つのクライアント公開鍵証明書が記憶された状態となる。この状態で通信相手に対して公開鍵証明書を送信する場合には、まず新公開鍵証明書を送信するものとする。ここではまだサーバ装置３０に新ルート鍵を記憶させていないので、サーバ装置３０は新公開鍵証明書のデジタル署名を復号化できず、認証が失敗した旨の応答を受けることになる。しかしこの場合でも、再度通信を要求し、この際に従前の公開鍵証明書を送信すれば、従前のルート鍵によってそこに付されたデジタル署名を復号化できるので、問題なく認証を受けることができる。

【０１２１】

なお、ステップＳ３１９及びＳ３２０の処理を、ステップＳ３１７及びＳ３１８の処理より前に行うようにしてもよい。この場合には、ステップＳ３１９における正当性の確認は、配布用ルート鍵証明書を用いて行うことになる。

クライアント装置４０はその後、ステップＳ３２１で、証明書管理装置１０に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置３０に対して送信し、サーバ装置３０がステップＳ３２２で証明書管理装置１０に対して送信する。

【０１２２】

その後、続いて図２２のシーケンス図に示す処理２２を行う。この処理は、第１の実施形態の説明において図５に示した処理１及び図８に示した処理４を併せ、さらに図９に示した処理５の一部を加えた処理に相当する。

ここではまず、ステップＳ３２３で、図８のステップＳ１４１の場合と同様に、証明書管理装置１０がサーバ公開鍵に新ルート私有鍵を用いたデジタル署名を付して新サーバ公開鍵証明書を作成する。

【０１２３】

そして、ステップＳ３２４で、証明書管理装置１０がサーバ装置３０に対して、図２０のステップＳ３０２で作成した配布用ルート鍵証明書と、図２０のステップＳ３０３で作成した新ルート鍵証明書と、ステップＳ３２３で作成した新サーバ公開鍵証明書と共に、これらについての更新要求を送信する。このステップＳ３２４の処理においては、証明書管理装置１０のＣＰＵ１１が第２の更新要求手段として機能する。

サーバ装置３０は、この要求を受け取ると、ステップＳ３２５及びＳ３２６で、図５のステップＳ１１２及びＳ１１３の場合と同様に、従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認し、これが確認できると配布用ルート鍵証明書を証明書記憶部３１に記憶する。このとき、まだ従前のルート鍵証明書は消去しない。

【０１２４】

そしてさらにステップＳ３２７で、図９のステップＳ１５３の場合と同様に、記憶した配布用ルート鍵証明書を用いて新ルート鍵証明書の正当性を確認する。そして、これが確認できると、次のステップＳ３２８で新ルート鍵証明書を証明書記憶部３１に記憶すると共に、配布用ルート鍵証明書と従前のルート鍵証明書を廃棄する。この時点では既にクライアント装置４０に新クライアント公開鍵証明書を記憶させてあるので、従前のルート鍵は不要であり、改めて廃棄要求を行うよりもここで廃棄してしまった方が処理の手順が簡単になるので、このようにしたものである。もちろん、改めて廃棄要求を行うようにしてもよい。

これらのステップＳ３２４乃至Ｓ３２８において、サーバ装置３０のＣＰＵが第２のサーバ側更新手段として機能する。

【０１２５】

次に、ステップＳ３２９及びＳ３３０で、図８のステップＳ１４３及びＳ１４４の場合と同様に、新サーバ公開鍵証明書の正当性を確認し、これが確認できると、新サーバ公開鍵証明書を証明書記憶部３１に記憶し、従前のサーバ公開鍵証明書と置き換える。ただし、ここでは既に新ルート鍵証明書を記憶しているので、新クライアント公開鍵証明書の正当性は、配布用ルート鍵証明書ではなく新ル

ート鍵証明書を用いて行うことができる。これらのステップＳ３２９及びＳ３３０において、サーバ装置３０のＣＰＵが第１のサーバ側更新手段として機能する。

【０１２６】

このとき従前のサーバ公開鍵証明書を消去する理由は、第１の実施形態において図７の説明で述べた通りである。そして、ステップＳ３３０の時点では既にクライアント装置に新ルート鍵を記憶させてあるので、新サーバ公開鍵証明書を記憶させておけば、認証処理には全く問題ない。

なお、ステップＳ３２９及びＳ３３０の処理を、ステップＳ３２７及びＳ３２８の処理より前に行うようにしてもよい。この場合には、ステップＳ３２９における正当性の確認は、配布用ルート鍵証明書を用いて行うことになる。

【０１２７】

サーバ装置３０はその後、ステップＳ３３１で証明書管理装置１０に対して更新要求に対する応答として結果通知を返す。

以上の図２２に示す処理により、サーバ装置３０側ではルート鍵更新処理が完了する。

【０１２８】

その後、続いて図２３のシーケンス図に示す処理２３を行う。

ここではまずステップＳ３３２で、証明書管理装置１０がサーバ装置３０に対して、不要になったデジタル証明書の廃棄を求める旧鍵廃棄要求をクライアント装置４０に送信するよう要求する旧鍵廃棄要求送信要求を送信する。サーバ装置３０は、これに応じて、クライアント装置４０からのポーリング（Ｓ３３３）に対する応答として旧鍵廃棄要求を送信するようにしている（Ｓ３３４）。

以上の処理により、証明書管理装置１０からクライアント装置４０にサーバ装置３０を介して上記の旧鍵廃棄要求が送信されることになる。

【０１２９】

クライアント装置４０は、この要求を受け取ると、ステップＳ３３５で、証明書記憶部４１に記憶している配布用ルート鍵証明書、従前のルート鍵証明書、および従前のクライアント公開鍵証明書を廃棄する。この時点では、サーバ装置３

０に新ルート鍵証明書及び新サーバ公開鍵証明書が記憶されているので、これらの証明書を消去しても相互認証に影響はない。

クライアント装置４０はその後、ステップＳ３３６で証明書管理装置１０に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置３０に対して送信し、サーバ装置３０がステップＳ３３７で証明書管理装置１０に対して送信する。

以上により、ルート鍵更新処理を終了する。

【０１３０】

このデジタル証明書管理システムにおいても、ルート鍵更新処理をこのような手順で行うことにより、第１の実施形態の場合と同様に、サーバ装置３０とクライアント装置４０との間の相互認証処理に大きな影響を与えることなく、ルート鍵を自動制御で更新することができる。従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおけるＳＳＬによる相互認証を、低コストで運用することができる。

【０１３１】

なお、この実施形態では、サーバ装置３０に新ルート鍵を記憶させる前にクライアント装置４０に新クライアント公開鍵証明書を記憶させるので、サーバ装置３０に新ルート鍵を記憶させるまでは、通信に、新クライアント公開鍵証明書のデジタル署名をサーバ装置３０が復号化できないことによるオーバーヘッドが生じる。しかし一方で、証明書管理装置１０からサーバ装置３０（あるいはサーバ装置３０を介してクライアント装置４０）に計３回の要求を送信するのみでルート鍵の更新処理を行うことができる。従って、６回の要求送信が必要な第１の実施形態の場合と比較して、処理手順の管理やプログラムの設計が容易であるという効果がある。ルート鍵証明書を更新すべきサーバ装置やクライアント装置の数が多い場合には、この効果はより大きくなり、この実施形態が有効である。

また、処理２１や処理２２において、各証明書について正当性を確認した後で必要なものを一括して記憶するようにすれば、証明書を記憶する不揮発メモリへのアクセス回数を低減し、処理負荷を低減すると共に処理を高速化することができる。

【０１３２】

〔第４の実施形態：図２０，図２４乃至図２６〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第４の実施形態の構成について説明する。

このデジタル証明書管理システムは、ルート鍵更新処理の内容が第２の実施形態のデジタル証明書管理システムと異なるのみであり、装置の構成は第２の実施形態のものと同様であるのでその説明は省略する。

【０１３３】

このデジタル証明書管理システムにおけるルート鍵更新動作は、この発明のデジタル証明書管理方法の第４の実施形態に係る動作であり、図２０及び図２４乃至図２６のシーケンス図に示す処理Ｔ及び処理３１乃至３３を、この順で実行するものである。そしてこれらの処理は、証明書管理装置１０，サーバ装置３０，クライアント装置４０の各ＣＰＵが、所要の制御プログラムを実行することによって行うものである。

【０１３４】

また、これらの処理は、図２０に示す部分については第３の実施形態の場合と共通であり、図２４乃至図２６に示す部分については、第３の実施形態で図２１乃至図２３を用いてそれぞれ説明した処理と同じ目的の処理であり、証明書管理装置１０と直接通信する装置がクライアント装置４０であることに伴って、第２の実施形態で図１３及び図１４を用いて説明した処理１１及び処理１２の場合と同様に通信手順を若干変更したのみである。そこで、これらの処理についての詳細な説明は省略する。

【０１３５】

そして、この第４の実施形態のデジタル証明書管理システムにおいても、ルート鍵更新処理をこのような手順で行うことにより、証明書管理装置１０がクライアント・サーバシステムを構成する装置のうちクライアント装置４０のみと通信可能な場合でも、第３の実施形態の場合と同様に、サーバ装置３０とクライアン

ト装置４０との間の相互認証処理に大きな影響を与えることなくルート鍵を自動制御で更新することができる。従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおけるＳＳＬによる相互認証を、低コストで運用することができる。また、処理手順の管理やプログラムの設計が容易であるという効果もある。

【０１３６】

〔第５の実施形態：図２７乃至図３１〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第５の実施形態の構成について説明する。

このデジタル証明書管理システムにおいては、図２７に示すように、クライアント・サーバシステムを１つのサーバ装置と複数のクライアント装置とによって構成している。個々の証明書管理装置１０，サーバ装置３０，クライアント装置４０の構成は第１の実施形態の場合と同様であるので、詳細な図示及び説明は省略するが、複数のクライアント装置４０－１～ｎを、サーバ装置３０と通信可能なように設けている。そして、証明書管理装置１０と各クライアント装置４０との通信は、サーバ装置３０が仲介して行う。

【０１３７】

ところで、このデジタル証明書管理システムにおいては、証明書管理装置１０の構成記憶部２６に、クライアント・サーバシステムを構成する各ノードの情報を、図２８に示す形式で記憶している。すなわち、各ノード毎に、ノードＩＤ、デジタル証明書管理装置（ＣＡ）１０との直接通信の可否、および、そのノードの通信相手となる各ノードのＩＤと共に、その通信相手と通信する際にクライアントとサーバのいずれとして機能するかを示す情報、その通信相手と通信する際に使用するルート鍵の情報、その通信相手におけるルート鍵証明書及び公開鍵証明書の更新状態を示す情報を記憶している。ここで、「通信相手」とは、認証を行った上で通信を行う相手を指すものとする。また、図示は省略したが、各ノードの情報として、そのノードが保有しているルート鍵証明書や公開鍵証明書のＩ

Ｄを、その有効期限と共に記憶するようにしてもよい。

【０１３８】

図２８に示した形式で記憶する具体的な情報としては、例えば図２７に示すサーバ装置３０については、図２９（ａ）に示すような情報を記憶することになる。すなわち、ノードＩＤとして「サーバ装置３０」を記憶し、証明書管理装置１０と直接通信可能であるのでその旨の情報を記憶している。そして、通信相手となるノードの情報として、クライアント装置４０－１～ｎの情報をそれぞれ記憶している。また、これらの各装置と通信する際に、サーバ装置３０はサーバとして機能するのでその旨を記憶し、使用するルート鍵の情報としてはここでは「ルート鍵Ａ」を記憶している。そして、ルート鍵Ａは更新が必要な状態であるとし、その旨の情報も記憶している。

【０１３９】

各クライアント装置４０－１～ｎについては、図２９（ｂ）と（ｃ）の双方の記録形態が考えられる。図にはクライアント装置４０－１についての記憶例を示しているが、ノードＩＤとして「クライアント装置４０－１」を記憶し、証明書管理装置１０とはサーバ装置３０を介して通信するので直接通信不能である旨の情報を記憶する点は、どちらも共通であるが、通信相手となるノードの情報の記録形態が異なる。

【０１４０】

すなわち、（ｂ）の形態では、サーバ装置３０とクライアント装置４０－１とが通信可能であることは、サーバ装置３０に関する情報として（ａ）に記録済であり、サーバ／クライアントの別等もその情報から導き出せるので、クライアント装置４０－１に関する情報として新たに記憶することはしていない。一方（ｃ）の形態では、クライアント装置４０－１に関する情報として別途サーバ装置３０とクライアント装置４０－１とが通信可能であることを記憶するようにしている。

（ｂ）の形態では情報の記憶容量が少なくて済み、（ｃ）の形態では対象ノードについての情報のみを参照すればそのノードの通信相手を知ることができる。しかし、どちらの形態を取るにせよ、各ノードの通信相手及びその通信相手との

間でクライアントとサーバのいずれとして機能するかの情報は記憶できているので、これを参照して後述のように証明鍵の更新手順を定めることができる。

【０１４１】

次に、図２７に示した第５の実施形態のデジタル証明書管理システムにおけるルート鍵更新処理について説明する。この処理は、この発明のデジタル証明書管理方法の第５の実施形態に係る処理である。

この処理は、基本的には、第１の実施形態で説明した処理Ｓ及び処理１乃至６を図３１のフローチャートに示す順番で実行するものである。そしてこれらの処理は、証明書管理装置１０，サーバ装置３０，クライアント装置４０－１～ｎの各ＣＰＵが、所要の制御プログラムを実行することによって行うものである。

ただし、この実施形態においては、クライアント装置４０を複数設けているので、これに伴ってクライアント装置４０に対して行う処理が若干異なったものになる。すなわち、各クライアント装置４０毎に、個別に配布用ルート鍵証明書や新クライアント証明書、新ルート鍵証明書を送信して記憶させるようにする必要があるのである。

【０１４２】

図３０に、図７に示したクライアント装置の公開鍵証明書記憶処理をクライアント装置４０－１に対して行う場合の処理シーケンスを処理３－１として示す。この図からわかるように、処理の流れ自体は図７に示した処理と変わらない。図３０に示した各処理は、図７に示した処理のうちステップ番号の下２ケタが一致する処理と対応するものである。しかし、ステップＳ５３１で作成する新クライアント公開鍵証明書は、クライアント装置４０－１が用いるためのものであり、ステップＳ５３２における更新要求送信要求においても、更新要求の送信先としてクライアント装置４０－１を指定している。

【０１４３】

このような処理は、当然他のクライアント装置４０－２～ｎについても行うが、実行時期についての条件が満たされていれば、初めの装置についての公開鍵証明書記憶処理に対する応答が帰ってくる前に次の装置についての公開鍵証明書記憶処理を行っても問題ない。また、複数の装置についての公開鍵証明書記憶処理

をまとめ、ステップＳ５３２でそれらの各装置に対応する新クライアント公開鍵証明書と各更新要求の送信先とを１つのメッセージに含める形でサーバ装置３０に送信するようにしてもよい。この場合でもステップＳ５３３乃至Ｓ５３７の処理をクライアント装置毎に行うことはもちろんであるが、ステップＳ５３８の結果通知については、クライアント装置毎に送信するようにしても、複数の装置からの結果通知を１つのメッセージに含める形で送信するようにしてもよい。

【０１４４】

なお、ここでは処理３に関する相違点について説明したが、図６に示した処理２及び図１０に示した処理６についても同様な点が第１の実施形態の場合と異なる。サーバ装置３０については、１つしか設けていないので、サーバ装置３０に対して行う処理１，４，５は第１の実施形態の場合と同様である。

また、上記の処理３－１という番号は、クライアント装置４０－１に対する処理３に相当する処理という意味で付したものであり、以下の説明においても、処理の番号は装置に付した符号の添え字を用いて同様に付すものとする。例えば、クライアント装置４０－ｎに対する処理３に相当する処理は処理３－ｎ，クライアント装置４０－１に対する処理６に相当する処理は処理６－１等である。

【０１４５】

このデジタル証明書管理システムにおける各処理の実行タイミングは、図３１に示すフローチャートのようなものになる。すなわち、ルート鍵の更新を行う場合には、まず図４に示した処理Ｓを実行し、その後処理１乃至処理６を実行する。

図３１の記載から明らかなように、この第５の実施形態におけるルート鍵更新処理は、図１１に示した第１の実施形態の場合の実行タイミングと概ね同様なものである。しかし、処理２，３，６を各クライアント装置に対して行う必要があることに伴い、若干異なったものになっている。

【０１４６】

具体的には、処理１及び処理２－１～ｎは処理Ｓの完了後に開始する。処理３－１～ｎは、処理２－１～ｎのうち対応する処理の完了後に開始する（例えば、処理３－１は処理２－１の完了後に開始する）が、処理１も完了した後に開始す

る方が好ましい。処理４は、処理１及び処理２－１～ｎの全てが完了した後に開始する。処理５は、処理１及び処理３－１～ｎの全てが完了した後に開始する。処理６は、処理２－１～ｎのうち対応する処理及び処理４の完了後に開始する。そして、処理３－１～ｎ，処理４，処理５，処理６－１～ｎが全て完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

なお、処理２，４，６については、それぞれの開始条件が満たされれば、各クライアント装置に対する処理は任意の順番で行って構わない。

【０１４７】

図３１に示す処理手順において、この実施形態の特徴となるのは、まず、処理４（サーバ装置の公開鍵証明書記憶処理）を、全てのクライアント装置４０について処理２（クライアント装置のルート鍵証明書記憶処理）が完了した後で、すなわちサーバ装置３０の通信相手となる全てのクライアント装置４０から配布用ルート鍵証明書を記憶した旨の応答があった後で実行する点である。

第１の実施形態で説明したように、サーバ装置３０については新サーバ公開鍵証明書を記憶させる際に従前のものを廃棄する必要があるので、通信相手となる全てのクライアント装置４０に新ルート鍵を記憶させる前にこれを行ってしまうと、認証処理に支障が生じるためである。逆に言えば、全てのクライアント装置４０に新ルート鍵を記憶させた後であれば、サーバ装置３０の従前のサーバ公開鍵証明書を廃棄してしまっても、認証処理に支障が生じることがない。

【０１４８】

また、処理３－１～ｎ（クライアント装置の公開鍵証明書記憶処理）を、処理１（サーバ装置のルート鍵証明書記憶処理）の後で、すなわち各クライアント装置４０－１～ｎについて、その通信相手となる全てのサーバ装置３０（ここでは１つだけ）から配布用ルート鍵証明書を記憶した旨の応答があった後で実行するようにするとよい。第１の実施形態で説明したように、クライアント装置４０に新クライアント公開鍵証明書を記憶させた時点で通信相手となるサーバ装置３０に新ルート鍵が記憶されていないと、そのサーバ装置３０に新ルート鍵が記憶されるまで通信にオーバーヘッドが生じ、効率が悪くなってしまうためである。

【０１４９】

その他の点も、クライアント装置４０を複数設けたことに伴って若干異なるが、概ね第１の実施形態の場合と同様であり、ルート鍵更新処理をこのような手順で行うことにより、第１の実施形態の場合と同様に、サーバ装置３０と各クライアント装置４０－１～ｎとの間の相互認証処理に大きな影響を与えることなく、ルート鍵を自動制御で更新することができる。

従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおけるＳＳＬによる相互認証を、低コストで運用することができる。

ここで、処理５（サーバ装置のルート鍵証明書置き換え処理）を、処理３－１～ｎの後で、すなわちサーバ装置３０の通信相手となる全てのクライアント装置４０－１～ｎから新クライアント公開鍵証明書を記憶した旨の応答があった後で実行するようにするとよい。また、処理６－１～ｎ（クライアント装置のルート鍵証明書置き換え処理）を、処理４の後で、すなわち各クライアント装置４０－１～ｎについて、その通信相手となる全てのサーバ装置３０（ここでは１つだけ）から新サーバ公開鍵証明書を記憶した旨の応答があった後で実行するようにするとよい。

【０１５０】

なお、図３１に示したような更新手順は、証明書管理装置１０の更新順制御部２７が構成記憶部２６に記憶している情報をもとに作成して管理する。そして、この更新手順の作成は、この発明の更新手順決定方法に係る処理である。この実施形態の場合には、まず証明書管理装置１０と直接通信可能なサーバ装置３０に関する情報を参照すると、このサーバ装置３０はサーバとして機能し、これと通信可能なノードとしてはクライアント装置４０－１～ｎがあることがわかる。そして、全てのノードとの通信に同じルート鍵Ａを使用し、更新が必要であることもわかる。さらに、各クライアント装置４０－１～ｎに関する情報を参照すると、このクライアント・サーバシステムにはそれ以上ノードがないことがわかるので、これらの情報から更新手順を作成することができる。

【０１５１】

すなわち、まずサーバ装置３０及びクライアント装置４０－１～ｎに配布用ル

ート鍵証明書を記憶させ、これらが全て終了したらサーバ装置３０に新サーバ公開鍵証明書を記憶させ、・・・、といったように、図３１に示した条件を満たすように更新に必要な各処理の実行順序を定めればよいのである。あるいは、処理４の実行には処理１及び処理２－１～ｎの全てが完了していることが必要等、各処理について実行条件を定め、これが満たされた場合にその処理を開始するように制御しても、更新手順を定めることができる。

【０１５２】

〔第５の実施形態の変形例：図３２乃至図３４〕

以上説明した第５の実施形態では、ルート鍵更新処理を図３１に示す手順で行う例について説明した。この処理手順は、必要最低限の条件のみを定めたものであるが、この条件のみに従うとすると、各処理の実行順序の決定や処理の進行状況の管理に当たって管理すべき情報が多くなる。そこで、ルート鍵更新処理を図３２又は図３３に示す手順で行うようにしてもよい。これらの図における矢印の意味は、図３１の場合と同様である。

【０１５３】

まず、図３２に示す例では、処理１及び処理２－１～ｎは処理Ｓの完了後に開始し、処理３－１～ｎは、これらの全ての処理の完了後に開始する。処理４は、処理３－１～ｎの全てが完了した後に開始する。そして、処理５及び処理６－１～ｎは、処理４の完了後に開始する。そして、処理５及び処理６－１～ｎが全て完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

このようにすれば、処理５や処理６の実行に当たって処理１や処理２の実行状況を監視する必要がない。処理４が完了している場合には、この処理の実行条件から、処理１と処理２も共に完了していることが保証されるためである。また、処理４の実行に当たっても、処理３－１～ｎのみの完了を確認すればよい。従って、処理の進行状況の管理を単純化することができる。

各処理の実行順序を決定する際にも、全てのノードに配布用新ルート鍵証明書を記憶させてから、クライアント装置→サーバ装置の順で新公開鍵証明書を記憶させるように定めればよいので、処理を単純化し、装置やプログラムの開発コストを低減することができる。

【０１５４】

また、図３３に示す例では、処理Ｓ，処理１，処理２－１～ｎ，処理３－１～ｎ，処理４をこの順で実行し、処理５及び処理６－１～ｎは、処理４の完了後に開始する。そして、処理５及び処理６－１～ｎが全て完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

このようにすれば、図３２に示した例の場合よりも、さらに処理の単純化を図ることができる。

一方で、図３２及び図３３に示した手順であっても、図３１に示した実行順序の条件は破線で示したものも含めて全て満たしているので、クライアント・サーバシステムにおける相互認証機能の維持という観点では、上述した第５の実施形態と同等の効果を有する。

【０１５５】

なお、図３３に示す手順に従う場合、処理２と処理３について、同一のクライアント装置４０に対して行う処理をまとめて行うことができる。この場合の処理例を図３４のシーケンス図に示す。ここでは、クライアント装置４０－１に対してルート鍵証明書記憶処理と公開鍵証明書記憶処理とをまとめて行う場合の処理を処理２′－１として示している。

この処理においては、証明書管理装置１０がステップＳ６２１で、図３０のステップＳ５３１の場合と同様にクライアント４０－１のための新クライアント証明書を作成する。そして、ステップＳ６２２で証明書管理装置１０がサーバ装置３０に対して、図４に示す処理Ｓのステップ１０２で作成した配布用ルート鍵証明書と、ステップＳ６２１で作成した新クライアント公開鍵証明書と共に、これらについての更新要求をクライアント装置４０－１に送信するよう要求する更新要求送信要求を送信する。

【０１５６】

サーバ装置３０はこれに応じて、図６のステップＳ１２２及びＳ１２３の場合と同様に、クライアント装置４０－１からのポーリング（Ｓ６２３）に対する応答としてこれらの証明書とそれらについての更新要求とを送信するようにしている（Ｓ６２４）。

これらの処理により、証明書管理装置１０からクライアント装置４０－１にサーバ装置３０を介して上記の各証明書とそれらについての更新要求とが送信されることになり、ステップＳ６２２の処理においては、証明書管理装置１０のＣＰＵ１１が第１の更新要求手段として機能する。

クライアント装置４０は、この要求を受け取ると、ステップＳ６２５及びＳ６２６で、図６のステップＳ１２４及びＳ１２５の場合と同様に、従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認し、これが確認できると配布用ルート鍵証明書を証明書記憶部４１に記憶する。このとき、まだ従前のルート鍵証明書は消去しない。これらの処理において、クライアント装置４０－１のＣＰＵが第２のクライアント側更新手段として機能する。

【０１５７】

次に、ステップＳ６２７及びＳ６２８で、図７のステップＳ１３５及びＳ１３６の場合と同様に、配布用ルート鍵証明書を用いて新クライアント公開鍵証明書の正当性を確認し、これが確認できると、新クライアント公開鍵証明書を証明書記憶部４１に記憶する。このとき、まだ従前のクライアント公開鍵証明書は消去しない。これらの処理において、クライアント装置４０－１のＣＰＵが第１のクライアント側更新手段として機能する。

クライアント装置４０－１はその後、ステップＳ６２９で、証明書管理装置１０に対して更新要求に対する応答として結果通知を返すが、これはまずサーバ装置３０に対して送信し、サーバ装置３０がステップＳ６３０で証明書管理装置１０に対して送信する。

【０１５８】

以上のように各クライアント装置４０について処理２と処理３をまとめて処理２′を行うことにより、第３の実施形態の場合のように、処理手順の管理やプログラムの設計が容易であるという効果がある。ここでは、クライアント装置側の処理についてはまとめていないので、この効果は後述する第７の実施形態の場合よりは小さいが、サーバ装置に新ルート鍵を記憶させてからクライアント装置４０に新クライアント公開鍵証明書を記憶させているので、通信のオーバーヘッドは生じないようにすることができる。

【０１５９】

〔第６の実施形態：図３５乃至図３８〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第６の実施形態の構成について説明する。

このデジタル証明書管理システムにおいては、図３５に示すように、クライアント・サーバシステムを１つのクライアント装置と複数のサーバ装置とによって構成している。個々の証明書管理装置１０，サーバ装置３０，クライアント装置４０の構成は第２の実施形態の場合と同様であるので、詳細な図示及び説明は省略するが、複数のサーバ装置３０－１～ｎを、クライアント装置４０の通信相手となるように設けている。そして、証明書管理装置１０と各サーバ装置３０との通信は、クライアント装置４０が仲介して行う。

【０１６０】

このようにクライアント・サーバシステムを構成した場合には、証明書管理装置１０の構成記憶部２６に記憶する、クライアント・サーバシステムを構成する各ノードの情報は、図３６に示すようなものになる。

すなわち、まずクライアント装置４０について、図３６（ａ）に示すような情報を記憶することになる。ここでは、ノードＩＤとして「クライアント装置４０」を記憶し、証明書管理装置１０と直接通信可能であるのでその旨の情報を記憶している。そして、通信相手となるノードの情報として、サーバ装置３０－１～ｎの情報をそれぞれ記憶している。また、これらの各装置と通信する際に、クライアント装置４０はクライアントとして機能するのでその旨を記憶し、使用するルート鍵の情報としてはここでは「ルート鍵Ａ」を記憶している。そして、ルート鍵Ａは更新が必要な状態であるとし、その旨の情報も記憶している。

【０１６１】

各サーバ装置３０－１～ｎについて、図３６の（ｂ）と（ｃ）の双方の記録形態が考えられることは第５の実施形態の場合と同様である。図にはサーバ装置３０－１についての記憶例を示しているが、ノードＩＤとして「サーバ装置３０－

１」を記憶し、証明書管理装置１０とはクライアント装置４０を介して通信するので直接通信不能である旨の情報を記憶している。

そして、こらの情報を参照して証明書管理装置１０の更新順制御部２７が証明鍵の更新手順を定めることができる。

なお、クライアント装置４０の場合には、通信相手のサーバ装置毎に認証処理に用いるルート鍵が異なる場合が考えられるが、この場合には、共通のルート鍵を使用するグループ毎に更新処理を行うものとする。すなわち、後述するものも含め、第１乃至第８の実施形態及びそれらの変形例をグループ毎に適用することにより、グループ毎に独立して更新処理を行うことができる。

【０１６２】

次に、図３５に示した第６の実施形態のデジタル証明書管理システムにおけるルート鍵更新処理について説明する。この処理は、この発明のデジタル証明書管理方法の第６の実施形態に係る処理である。

この処理は、基本的には、第２の実施形態で説明した処理Ｓ及び処理１１乃至１６を図３８のフローチャートに示す順番で実行するものである。そしてこれらの処理は、証明書管理装置１０，サーバ装置３０－１～ｎ，クライアント装置４０の各ＣＰＵが、所要の制御プログラムを実行することによって行うものである。

ただし、この実施形態においては、サーバ装置３０を複数設けているので、これに伴ってサーバ装置３０に対して行う処理が若干異なったものになる。すなわち、各サーバ装置３０毎に、個別に配布用ルート鍵証明書や新クライアント証明書、新ルート鍵証明書を送信して記憶させるようにする必要があるのである。

【０１６３】

図３７に、図１６に示したサーバ装置の公開鍵証明書記憶処理をサーバ装置３０－１に対して行う場合の処理シーケンスを処理１４－１として示す。この図からわかるように、処理の流れ自体は図１６に示した処理と変わらない。図３７に示した各処理は、図１６に示した処理のうちステップ番号の下２ケタが一致する処理と対応するものである。しかし、ステップＳ７４１で作成する新クライアント公開鍵証明書は、サーバ装置３０－１が用いるためのものであり、ステップＳ

７４２における更新要求送信要求においても、更新要求の送信先としてサーバ装置３０－１を指定している。

このように、図１５に示した処理１４と図３７に示した処理１４－１との対応関係は、第５の実施形態で説明した処理３と処理３－１との対応関係と同様なものであり、処理１４と比較した場合のその他の相違点も、第５の実施形態で処理３－１について説明したものと同様である。

【０１６４】

なお、図１３に示した処理１１及び図１７に示した処理１５についても同様な点が第２の実施形態の場合と異なる。クライアント装置４０は１つしか設けていないので、クライアント装置４０に対して行う処理１２，１３，１６は第２の実施形態の場合と同様である。

また、上記の処理１４－１という番号は、サーバ装置３０－１に対する処理１４に相当する処理という意味で付したものであり、以下の説明においても、処理の番号は装置に付した符号の添え字を用いて同様に付すものとする。

【０１６５】

このデジタル証明書管理システムにおける各処理の実行タイミングは、図３８に示すフローチャートのようなものになる。すなわち、ルート鍵の更新を行う場合には、まず図４に示した処理Ｓを実行し、その後処理１１乃至処理１６を実行する。

図３８の記載から明らかなように、この第６の実施形態におけるルート鍵更新処理は、図１９に示した第２の実施形態の場合の実行タイミングと概ね同様なものである。しかし、処理１１，１４，１５を各サーバ装置に対して行う必要があることに伴い、若干異なったものになっている。

【０１６６】

具体的には、処理１１－１～ｎ及び処理１２は処理Ｓの完了後に開始する。処理１３は、処理１２の完了後に開始するが、処理１１－１～ｎが全て完了した後に開始する方が好ましい。処理１４－１～ｎは、処理１２及び処理１１－１～ｎのうち対応する処理の完了後に開始する（例えば、処理１４－１は処理１１－１の完了後に開始する）。処理１５は、処理１１－１～ｎのうち対応する処理及び

処理１３の完了後に開始する。処理１６は、処理１２及び処理１４－１～ｎが全て完了した後に開始する。そして、処理１３，処理１４－１～ｎ，処理１５－１～ｎ，処理１６が全て完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

この処理は、証明書管理装置１０と直接通信する装置がクライアント装置４０であることと、クライアント装置４０でなくサーバ装置３０を複数設けたことに応じて若干の相違があるが、基本的には、図３１に示した第５の実施形態の場合と対応する処理を、同様な順序で行うものである。そして、このことによる効果も、第５の実施形態の場合と同様である。

【０１６７】

すなわち、この第６の実施形態のデジタル証明書管理システムにおいては、ルート鍵更新処理をこのような手順で行うことにより、証明書管理装置１０がクライアント・サーバシステムを構成する装置のうちクライアント装置４０のみと直接通信可能であり、サーバ装置を複数設けた場合でも、第５の実施形態の場合と同様に、サーバ装置３０とクライアント装置４０との間の相互認証処理に大きな影響を与えることなくルート鍵を自動制御で更新することができる。従って、このようなデジタル証明書管理システムを用いることにより、クライアント・サーバシステムにおけるＳＳＬによる相互認証を、低コストで運用することができる。

【０１６８】

また、ルート鍵更新処理の手順にポーリング待ちを必要とする箇所がないため、処理を速やかに進め、短期間で完了させることができることは、第２の実施形態の場合と同様である。

その他、第５の実施形態の場合と同様な変形を、この第６の実施形態に適用することも可能である。

【０１６９】

〔第７の実施形態：図３９〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによっ

て構成される、この発明のデジタル証明書管理システムの第７の実施形態の構成について説明する。

このデジタル証明書管理システムは、ルート鍵更新処理の内容が第５の実施形態のデジタル証明書管理システムと異なるのみであり、装置の構成は第５の実施形態のものと同様であるのでその説明は省略する。

【０１７０】

このデジタル証明書管理システムにおけるルート鍵更新動作は、この発明のデジタル証明書管理方法の第７の実施形態に係る動作である。そしてこの処理は、基本的には、第３の実施形態で説明した図２０に示す処理Ｔ及び図２１乃至図２３にそれぞれ示す処理２１乃至２３をこの順で実行するものである。そしてこれらの処理は、証明書管理装置１０，サーバ装置３０，クライアント装置４０－１～ｎの各ＣＰＵが、所要の制御プログラムを実行することによって行うものである。

ただし、この実施形態においては、クライアント装置４０を複数設けているので、これに伴ってクライアント装置４０に対して行う処理が若干異なったものになる。すなわち、第５の実施形態の場合と同様に、各クライアント装置４０毎に個別に配布用ルート鍵証明書や新クライアント証明書、新ルート鍵証明書を送信して記憶させるようにする必要があるのである。

【０１７１】

具体的には、図２１に示した処理２１と、図２３に示した処理２３とを、各クライアント装置毎に行う。これに伴う処理の変更内容及び処理の呼称は、第５の実施形態において説明した処理３と処理３－１との対応関係と同様であるので、図示は省略するが、例えば処理２１－１において図２１に示す処理２１のステップＳ３１１に相当する処理で作成する新クライアント公開鍵証明書は、クライアント装置４０－１が用いるためのものであり、ステップＳ３１２に相当する処理における更新要求送信要求においても、更新要求の送信先としてクライアント装置４０－１を指定する。

【０１７２】

そして、ルート鍵更新処理においては、各処理は図３９のフローチャートに示

すタイミングで行う。

すなわち、まず処理Ｔを開始し、その完了後に処理２１－１～ｎを任意の順番で開始する。これらの全てが完了した後で処理２２を開始し、その完了後に処理２３－１～ｎを任意の順番で開始する。そして、これらの全てが完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

【０１７３】

このような更新手順は、証明書管理装置１０の更新順制御部２７が構成記憶部２６に記憶している情報をもとに作成して管理する。そして、この更新手順の作成は、この発明の更新手順決定方法に係る処理である。この実施形態の場合には、各ノードに関する情報を参照すると、クライアント・サーバシステムにおいてクライアントとして機能するノードがクライアント装置４０－１～ｎであることがわかるので、まずこれらについて更新処理を行い、その完了後に、サーバとして機能するサーバ装置３０についての更新処理を行うように更新手順を定めればよい。そして、サーバ側の更新処理も終了した後で、クライアント側の旧鍵廃棄処理を行うようにすればよいのである。

以上のような手順で更新処理を行うことにより、第３の実施形態の場合と同様に一部通信のオーバーヘッドが生じるが、第５の実施形態の場合と比較して、処理手順の管理やプログラムの設計が容易であるという効果がある。ルート鍵証明書を更新すべきノードの数が多い場合には、この効果はより大きくなり、この実施形態が有効である。

【０１７４】

〔第８の実施形態：図４０〕

次に、この発明によるデジタル証明書管理装置である証明書管理装置と、クライアント・サーバシステムを構成するクライアント装置及びサーバ装置とによって構成される、この発明のデジタル証明書管理システムの第８の実施形態の構成について説明する。

このデジタル証明書管理システムは、ルート鍵更新処理の内容が第６の実施形態のデジタル証明書管理システムと異なるのみであり、装置の構成は第６の実施形態のものと同様であるのでその説明は省略する。

【０１７５】

このデジタル証明書管理システムにおけるルート鍵更新動作は、この発明のデジタル証明書管理方法の第８の実施形態に係る動作である。そしてこの処理は、基本的には、第４の実施形態で説明した処理Ｔ及び処理３１乃至３３をこの順で実行するものである。そしてこれらの処理は、証明書管理装置１０，サーバ装置３０－１～ｎ，クライアント装置４０の各ＣＰＵが、所要の制御プログラムを実行することによって行うものである。

ただし、この実施形態においては、サーバ装置３０を複数設けているので、これに伴ってサーバ装置３０に対して行う処理が若干異なったものになる。すなわち、第６の実施形態の場合と同様に、各サーバ装置３０毎に個別に配布用ルート鍵証明書や新クライアント証明書、新ルート鍵証明書を送信して記憶させるようにする必要があるのである。

【０１７６】

具体的には、図２５に示した処理３２を、各サーバ装置毎に行う。これに伴う処理の変更内容及び処理の呼称は、第６の実施形態において説明した処理１４と処理１４－１との対応関係と同様であるので、図示は省略するが、例えば処理３２－１において図２５に示す処理３２のステップＳ４２０に相当する処理で作成する新サーバ公開鍵証明書は、サーバ装置３０－１が用いるためのものであり、ステップＳ４２１に相当する処理における更新要求送信要求においても、更新要求の送信先としてサーバ装置３０－１を指定する。

そして、ルート鍵更新処理においては、各処理は図４０のフローチャートに示すタイミングで行う。

すなわち、まず処理Ｔを開始し、その完了後に処理３１を開始する。そして、この処理が完了した後で処理２２－１～ｎを任意の順番で開始し、その全てが完了した後に処理２３を開始する。この処理が完了した時点で、ルート鍵及び公開鍵証明書の更新が終了したことになる。

【０１７７】

このような更新手順は、証明書管理装置１０の更新順制御部２７が構成記憶部２６に記憶している情報をもとに作成して管理する。そして、この更新手順の作

成は、この発明の更新手順決定方法に係る処理である。この実施形態の場合には、各ノードに関する情報を参照すると、クライアント・サーバシステムにおいてクライアントとして機能するノードがクライアント装置４０であることがわかるので、まずこれについて更新処理を行い、その完了後に、このクライアント装置４０の通信相手であるサーバとして機能するサーバ装置３０－１～ｎについての更新処理を行うように更新手順を定めればよい。そして、サーバ側の更新処理も全て終了した後で、クライアント側の旧鍵廃棄処理を行うようにすればよいのである。

以上のような手順で更新処理を行うことにより、第４の実施形態の場合と同様に一部通信のオーバーヘッドが生じるが、第６の実施形態の場合と比較して、処理手順の管理やプログラムの設計が容易であるという効果がある。ルート鍵証明書を更新すべきノードの数が多い場合には、この効果はより大きくなり、この実施形態が有効である。

【０１７８】

〔第５乃至第８の実施形態の変形例：図４１乃至図４４〕

上述した第５乃至第８の実施形態では、証明書管理装置１０と直接通信するノード１つと、そのノードの通信相手となる複数のノードとによってクライアント・サーバシステムを構成した場合の例について説明した。しかし、この発明は、図４１及び図４２に示すように、クライアント・サーバシステムを構成するサーバとクライアントとをそれぞれ複数設け、これらのノードのうち、複数のノードを証明書管理装置１０と直接通信可能とする場合にも適用できる。

ここで、このような場合のルート鍵の更新手順について説明する。なお、図４１あるいは図４２に示したクライアント・サーバシステムにおいて、各ノード間の相互認証には全て同じルート鍵を使用するものとする。

【０１７９】

まず、図４１には、証明書管理装置１０と直接通信可能なサーバ装置３０を複数設けているが、全てのクライアント装置４０が１つのみのサーバ装置３０を通信相手とする場合の例を示している。このような場合には、サーバ装置毎に別々のクライアント・サーバシステムがあるものとして更新処理を行うことができる

。

すなわち、図４１に示した例では、サーバ装置３０－１及びクライアント装置４０－１～３で構成されたクライアント・サーバシステムと、サーバ装置３０－２及びクライアント装置４０－４～５で構成されたクライアント・サーバシステムとに対して独立にルート鍵更新処理を行うようにすればよい。システムをまたいだ認証処理は行われないのであるから、このようにしても、各クライアント・サーバシステムに対して第５あるいは第７の実施形態で説明した更新手順で更新処理を行うようにすれば、各ノードの間での認証処理に大きな支障を来すことなく、ルート鍵の更新を行うことができる。

【０１８０】

また、図４２には、複数のサーバ装置を通信相手とするクライアント装置（クライアント装置４０－３）が存在する場合の例を示している。このような場合には、全てのノードによって１つのクライアント・サーバシステムが構成されるものとして更新処理を行う必要がある。しかし、このような場合であっても、それぞれのサーバ装置に新サーバ証明書を記憶させる処理を、そのサーバ装置の通信相手となる全てのクライアント装置に対して新ルート鍵を記憶させた後で行うようにすればよいことは、第５及び第７の実施形態の場合と同様である。

【０１８１】

この例の場合の更新処理に必要な各処理の開始条件を図示すると、図４３のようになる。この図において、各矢印や処理番号の意味は、第５の実施形態の説明で用いた図３１の場合と同様である。クライアント・サーバシステムの構成が複雑になったことに伴い、開始条件の内容も図３１と比較してかなり複雑になる。しかし、例えばサーバ装置３０－１についての公開鍵証明書記憶処理である処理４－１は、そのサーバ装置３０－１自身及びその通信相手となるクライアント装置４０－１～３についてのルート鍵証明書記憶処理である処理１－１及び処理２－１～３が全て完了してから開始する等、各処理の開始条件は、図３１の場合と同様な規則に基づいている。クライアント装置４０－３についての公開鍵証明書記憶処理である処理３－３を、そのサーバ装置４０－３自身及びその通信相手となるサーバ装置３０－１，２についてのルート鍵証明書記憶処理である処理２－

３及び処理１－１，２が全て完了してから開始するようにするとよいことも、図３１の場合と同様な規則から導き出すことができる。

ただし、サーバ装置３０－１とクライアント装置４０－４等、互いに通信相手とならないノード間については、もともと認証処理は行わないのであるから、相互の証明書の記憶状況を適切な関係に保つ必要はなく、処理順序の管理も必ずしも行う必要はない。

【０１８２】

また、第７の実施形態の場合のように、各ノードにルート鍵証明書と公開鍵証明書とを一括して記憶させる場合には、各処理の開始条件は図４４のようになる。この図は図３９と対応するものであり、このような処理手順も、図３９の場合と同様に、サーバ装置に対する更新処理を、そのサーバ装置の通信相手となる全てのクライアント装置に対する更新処理が完了した後で開始するという条件に従って定めることができる。

このような図４３及び図４４に示したような更新手順も、第５あるいは第７の実施形態の場合と同様に、証明書管理装置１０の更新順制御部２７が構成記憶部２６に記憶している情報をもとに作成して管理する。クライアント・サーバシステムの構成が図４２に示すようなものであっても、構成記憶部２６に記憶している各ノードに関する情報を参照することにより、各ノードの通信相手及びその機能を把握することができるので、それを基に更新手順を作成することができるのである。

【０１８３】

ここで、更新手順を作成する場合において、クライアント装置４０－３のように複数のサーバ装置３０と通信可能なノードへの要求は、どちらのサーバ装置３０を介して行うようにしてもよい。

なお、ここで説明した変形例では、証明書管理装置１０と直接通信可能なノードがサーバ装置である例について説明したが、これがクライアント装置であっても同様な変形を適用できることはもちろんであり、この場合には、上述のような変形を第６あるいは第８の実施形態に適用することになる。

【０１８４】

〔上述した各実施形態についての他の変形例：図４５〕

以上説明した実施形態では、クライアント装置４０とサーバ装置３０とが図４６を用いて説明したようなＳＳＬによる相互認証を行う場合の例について説明した。しかし、この相互認証が必ずしもこのようなものでなくてもこの発明は効果を発揮する。

ＳＳＬを改良したＴＬＳ（Transport Layer Security）も知られているが、このプロトコルに基づく認証処理を行う場合にも当然適用可能である。

【０１８５】

また、上述した実施形態では、証明書管理装置１０をサーバ装置３０あるいはクライアント装置４０とは別に設ける例について説明したが、サーバ装置３０あるいはクライアント装置４０と一体として設けることを妨げるものではない。この場合、証明書管理装置１０の機能を実現するためのＣＰＵ，ＲＯＭ，ＲＡＭ等の部品を独立して設けてもよいが、ハードウェア資源としてはサーバ装置３０あるいはクライアント装置４０のＣＰＵ，ＲＯＭ，ＲＡＭ等を使用し、そのＣＰＵに適当なソフトウェアを実行させることにより、証明書管理装置１０として機能させるようにしてもよい。

【０１８６】

このような場合において、証明書管理装置１０と、これと一体になっているサーバ装置３０あるいはクライアント装置４０との間の通信には、ハードウェアを証明書管理装置１０として機能させるためのプロセスと、ハードウェアをサーバ装置３０あるいはクライアント装置４０として機能させるためのプロセスとの間のプロセス間通信を含むものとする。

さらに、上述した各実施形態では、証明書管理装置１０が証明鍵やデジタル証明書を自ら作成してこれを取得する例について説明したが、図２及び図１２に示した証明用鍵作成部２１や証明書発行部２２の機能を証明書管理装置１０とは別の装置に設け、証明書管理装置１０がその装置から証明鍵やデジタル証明書の供給を受けてこれらを取得するようにしてもよい。

【０１８７】

また、証明書管理装置１０がサーバ装置３０及びクライアント装置４０の双方

と直接的に通信が可能な構成としても構わない。この場合、図５乃至図１０等に示した通信シーケンスは、双方の装置と直接通信が可能であることに伴って異なったものになるが、処理の順序は上述した各実施形態の場合と同様である。このようにしても、上述した各実施形態の効果を得ることができる。

【０１８８】

また、上述したように、第２及び第４の実施形態においては、証明書管理装置１０とクライアント装置４０との間で通信を行う際にも、ＳＳＬによる相互認証を行うようにすることができる。

このようにするには、図４５に示すように、クライアント装置４０に、サーバ装置３０との相互認証に用いるクライアント秘密鍵、クライアント公開鍵証明書及びルート鍵証明書（実施形態において説明したもの）とは別に、もう一組の秘密鍵、公開鍵証明書及びルート鍵証明書（「第２のクライアント秘密鍵」、「第２のクライアント公開鍵証明書」及び「第２のルート鍵証明書」と呼ぶ）を記憶させ、証明書管理装置１０との相互認証にこれらを用いるようにすればよい。

【０１８９】

この場合、証明書管理装置１０にも、管理装置用秘密鍵、管理装置用公開鍵証明書及び上記の第２のルート鍵証明書を記憶させ、相互認証に用いる。そして、第２のクライアント公開鍵証明書及び管理装置用公開鍵証明書は、第２のルート鍵証明書に含まれる第２のルート鍵で内容が確認できるものとする。すなわち、その第２のルート鍵と対応するルート私有鍵（第２のルート私有鍵）を用いてデジタル署名を付すようにする。

このようにすれば、証明書管理装置１０とクライアント装置４０との間の相互認証と、クライアント装置４０とサーバ装置３０との間の相互認証とを、全く独立して行うことができる。

【０１９０】

第２及び第４の実施形態におけるクライアント装置４０は、図１２を用いて説明したように、証明書管理装置１０との通信はサーバ機能部４４が、サーバ装置３０との通信はクライアント機能部４３が通信機能部４２を介して行う。従って、証明書管理装置１０から通信を要求される通信と、サーバ装置３０に要求する

通信とは明確に区別することができるため、これらとの間で別々の鍵や証明書を用いた相互認証を行うことができるのである。

このような場合において、証明書管理装置１０からの要求に応じてクライアント装置４０とサーバ装置３０との間の相互認証に用いるルート鍵証明書や公開鍵証明書を更新したとしても、証明書管理装置１０とクライアント装置４０との間の相互認証には全く影響がない。

【０１９１】

各実施形態で説明した手順によって更新処理を行えば、クライアント装置４０とサーバ装置３０との間の相互認証にも大きな影響を与えることなく更新処理を行えることは上述した通りであるので、図４５に示した構成をとることにより、各ノード間の相互認証を維持したままルート鍵を更新できると言える。

なお、第２のルート鍵証明書を更新しようとする場合には、証明書管理装置１０をクライアント、クライアント装置４０をサーバとして、上述したいずれかの実施形態の手順に従って更新処理を行えばよい。このような更新処理を行っても、クライアント装置４０とサーバ装置３０との間の相互認証には全く影響がない。

第６及び第８の実施形態のように、サーバ装置３０を複数設けた場合であっても、同様な対応が可能である。

【０１９２】

また、この発明によるプログラムは、クライアント・サーバシステムを構成する複数の装置とネットワークを介して直接的又は間接的に通信可能なコンピュータに、この発明による各機能（構成記憶手段，更新順制御手段，証明鍵更新手段，第１の更新要求手段，第２の更新要求手段，その他の手段としての機能）を実現させるためのプログラムであり、このようなプログラムをコンピュータに実行させることにより、上述したような効果を得ることができる。

【０１９３】

このようなプログラムは、はじめからコンピュータに備えるＲＯＭあるいはＨＤＤ等の記憶手段に格納しておいてもよいが、記録媒体であるＣＤ－ＲＯＭあるいはフレキシブルディスク，ＳＲＡＭ，ＥＥＰＲＯＭ，メモリカード等の不揮発

性記録媒体（メモリ）に記録して提供することもできる。そのメモリに記録されたプログラムをコンピュータにインストールしてＣＰＵに実行させるか、ＣＰＵにそのメモリからこのプログラムを読み出して実行させることにより、上述した各手順を実行させることができる。

さらに、ネットワークに接続され、プログラムを記録した記録媒体を備える外部機器あるいはプログラムを記憶手段に記憶した外部機器からダウンロードして実行させることも可能である。

【０１９４】

【発明の効果】

以上説明してきた通り、この発明のデジタル証明書管理システム、デジタル証明書管理装置、デジタル証明書管理方法によれば、クライアント・サーバシステムにおける認証処理でデジタル証明書の正当性の確認に用いる証明鍵を、その認証処理に支障を来すことなく自動的に更新することができる。そして、このことにより、公開鍵暗号を利用したデジタル証明書を用いるＳＳＬ等の方式による相互認証を、クライアント・サーバシステムにおいて低コストで実現可能とすることができる。

この発明の更新手順決定方法によれば、上記のような証明鍵の更新を行うための更新処理の適切な手順を決定できるので、適当な更新装置にこの手順に従って更新処理を行わせることにより、同様な効果を得ることができる。

また、この発明のプログラムによれば、コンピュータにデジタル証明書管理装置を制御させてこのようなデジタル証明書管理装置の特徴を実現し、同様な効果を得ることができる。

【図面の簡単な説明】

【図１】

この発明のデジタル証明書管理装置の実施形態である証明書管理装置のハードウェア構成を示すブロック図である。

【図２】

この発明のデジタル証明書管理システムの第１の実施形態を構成する各装置の、この発明の特徴となる部分の機能構成を示す機能ブロック図である。

【図３】

図２に示したデジタル証明書管理システムにおけるデータ送受モデルを示す概念図である。

【図４】

図２に示したデジタル証明書管理システムにおけるルート鍵更新処理のうち、ルート鍵証明書作成処理を示すシーケンス図である。

【図５】

同じくサーバ装置のルート鍵証明書記憶処理を示すシーケンス図である。

【図６】

同じくクライアント装置のルート鍵証明書記憶処理を示すシーケンス図である。

【図７】

同じくクライアント装置の公開鍵証明書記憶処理を示すシーケンス図である。

【図８】

同じくサーバ装置の公開鍵証明書記憶処理を示すシーケンス図である。

【図９】

同じくサーバ装置のルート鍵証明書書き換え処理を示すシーケンス図である。

【図１０】

同じくクライアント装置のルート鍵証明書書き換え処理を示すシーケンス図である。

【図１１】

第１の実施形態のルート鍵更新処理における、図４乃至図１０のシーケンス図に示した各処理の実行順を示すフローチャートである。

【図１２】

この発明のデジタル証明書管理システムの第２の実施形態を構成する各装置の、この発明の特徴となる部分の機能構成を示す機能ブロック図である。

【図１３】

図１２に示したデジタル証明書管理システムにおけるルート鍵更新処理のうち、サーバ装置のルート鍵証明書記憶処理を示すシーケンス図である。

【図１４】

同じくクライアント装置のルート鍵証明書記憶処理を示すシーケンス図である。

【図１５】

同じくクライアント装置の公開鍵証明書記憶処理を示すシーケンス図である。

【図１６】

同じくサーバ装置の公開鍵証明書記憶処理を示すシーケンス図である。

【図１７】

同じくサーバ装置のルート鍵証明書書き換え処理を示すシーケンス図である。

【図１８】

同じくクライアント装置のルート鍵証明書書き換え処理を示すシーケンス図である。

【図１９】

第２の実施形態のルート鍵更新処理における、図４及び図１３乃至図１８のシーケンス図に示した各処理の実行順を示すフローチャートである。

【図２０】

この発明のデジタル証明書管理システムの第３の実施形態におけるルート鍵更新処理の一部を示すシーケンス図である。

【図２１】

図２０の続きの処理を示すシーケンス図である。

【図２２】

図２１の続きの処理を示すシーケンス図である。

【図２３】

図２２の続きの処理を示すシーケンス図である。

【図２４】

この発明のデジタル証明書管理システムの第４の実施形態におけるルート鍵更新処理の、図２０の続きの処理を示すシーケンス図である。

【図２５】

図２４の続きの処理を示すシーケンス図である。

【図２６】

図２５の続きの処理を示すシーケンス図である。

【図２７】

この発明のデジタル証明書管理システムの第５の実施形態を構成する各装置の関係を示すブロック図である。

【図２８】

図２に示した構成記憶部２６に記憶する各ノードの情報の記憶形式の例を示す図である。

【図２９】

図２７に示したサーバ装置３０及びクライアント装置４０－１について、図２８に示した形式で情報を記載した場合の記載例を示す図である。

【図３０】

第１の実施形態で説明した処理を第５の実施形態に適用する場合の変更点について説明するための図である。

【図３１】

第５の実施形態のルート鍵更新処理における、各処理の実行順を示す図１１と対応するフローチャートである。

【図３２】

その変形例のルート鍵更新処理における、各処理の実行順を示すフローチャートである。

【図３３】

その別の変形例のルート鍵更新処理における、各処理の実行順を示すフローチャートである。

【図３４】

図３３に示した処理において、クライアント装置のルート鍵証明書記憶処理と公開鍵証明書記憶処理とをまとめて行う場合の処理を示すシーケンス図である。

【図３５】

この発明のデジタル証明書管理システムの第６の実施形態を構成する各装置の関係を示すブロック図である。

【図３６】

図３５に示したクライアント装置４０及びサーバ装置３０－１について、図２８に示した形式で情報を記載した場合の記載例を示す図である。

【図３７】

第２の実施形態で説明した処理を第６の実施形態に適用する場合の変更点について説明するための図である。

【図３８】

第６の実施形態のルート鍵更新処理における、各処理の実行順を示す図１９と対応するフローチャートである。

【図３９】

第７の実施形態のルート鍵更新処理における、各処理の実行順を示すフローチャートである。

【図４０】

第８の実施形態のルート鍵更新処理における、各処理の実行順を示すフローチャートである。

【図４１】

この発明のデジタル証明書管理システムの第５乃至第８の実施形態に適用する変形例を構成する各装置の関係を示すブロック図である。

【図４２】

その別の変形例を構成する各装置の関係を示すブロック図である。

【図４３】

図４２に示した構成のデジタル証明書管理システムにおけるルート鍵更新処理を構成する各処理の開始条件を示す図である。

【図４４】

同じく、各ノードにルート鍵証明書と公開鍵証明書とを一括して記憶させる場合の各処理の開始条件を示す図である。

【図４５】

別の変形例における、鍵及び証明書の記憶状態及びその場合のルート鍵更新処理について説明するための図である。

【図４６】

クライアント装置とサーバ装置とがＳＳＬによる相互認証を行う際の各装置において実行する処理のフローチャートを、その処理に用いる情報と共に示す図である。

【図４７】

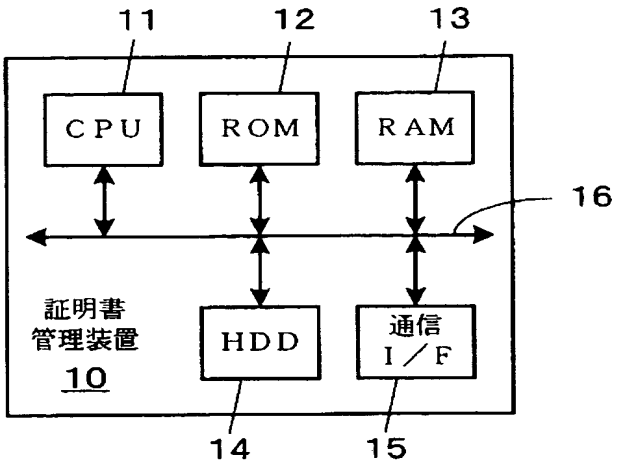
図４６に示した認証処理におけるルート鍵、ルート私有鍵、およびクライアント公開鍵の関係について説明するための図である。

【符号の説明】

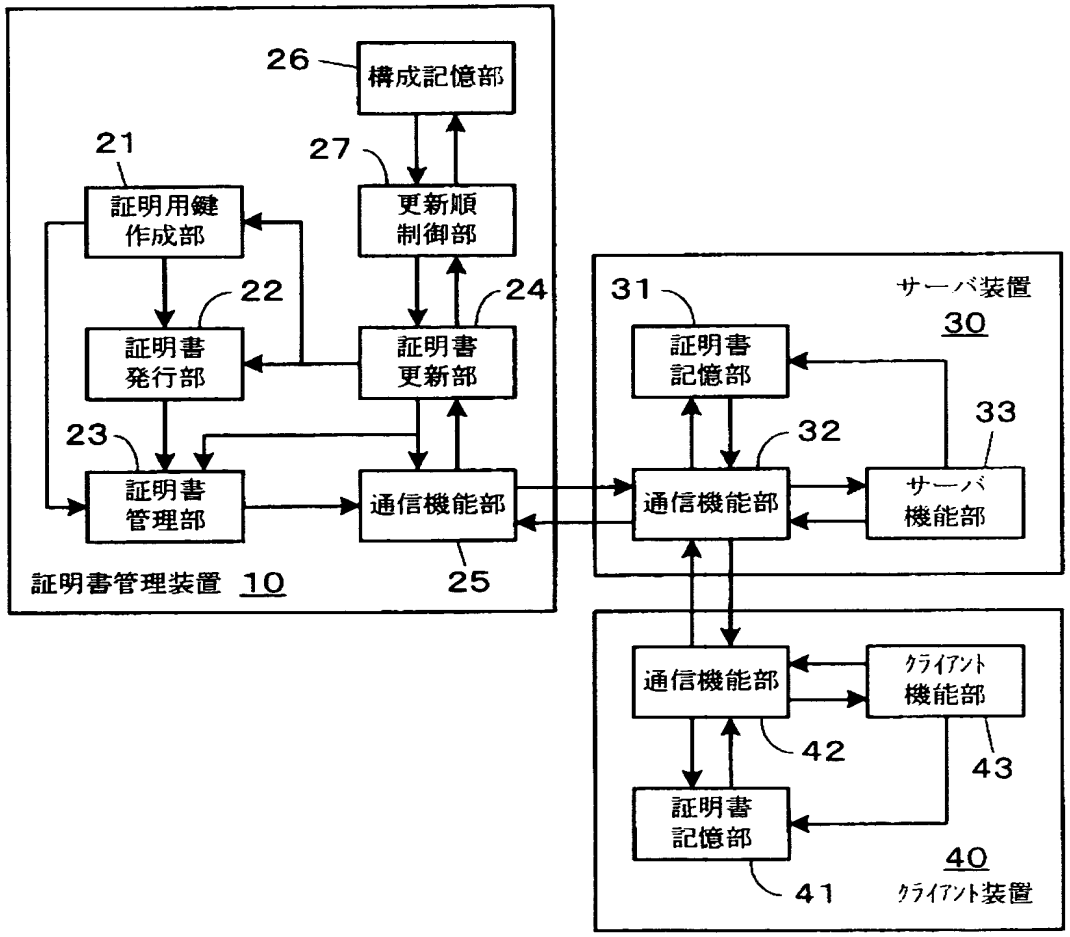
１０：証明書管理装置　　１１：ＣＰＵ
１２：ＲＯＭ　　１３：ＲＡＭ
１４：ＨＤＤ　　１５：通信Ｉ／Ｆ
１６：システムバス　　２１：証明用鍵作成部
２２：証明書発行部　　２３：証明書管理部
２４：証明書更新部　　２５，３２，４２：通信機能部
３０：サーバ装置　　３１，４１：証明書記憶部
３３，４４：サーバ機能部　　４０：クライアント装置
４３：クライアント機能部

【書類名】　図面

【図１】

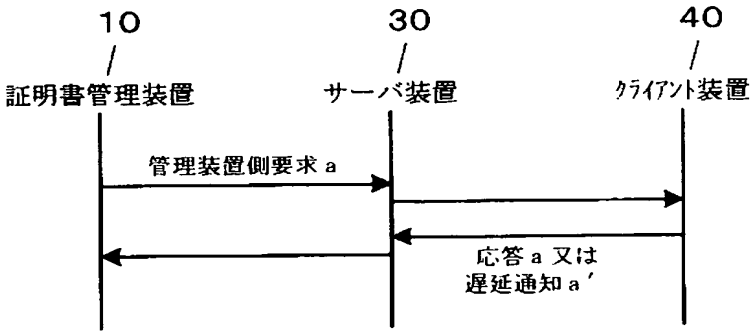

11
12
13
CPU
ROM
RAM
16
証明書
管理装置
10
HDD
通信
I／F
14
15

【図２】

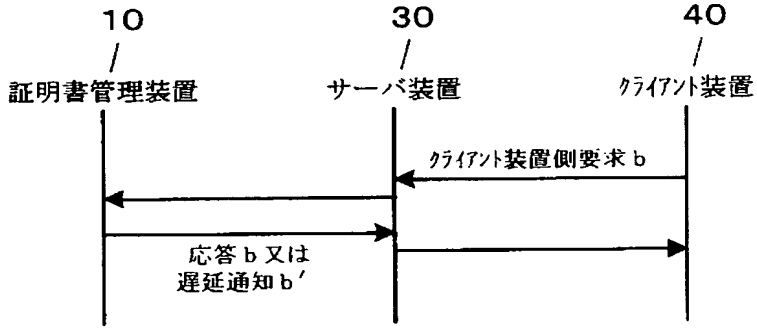

26
構成記憶部
21
27
証明用鍵
作成部
更新順
制御部
22
24
証明書
発行部
証明書
更新部
23
証明書
管理部
通信機能部
25
証明書管理装置　10
31
サーバ装置
30
証明書
記憶部
32
33
通信機能部
サーバ
機能部
通信機能部
ｸﾗｲｱﾝﾄ
機能部
42
43
証明書
記憶部
40
41
ｸﾗｲｱﾝﾄ装置

【図３】

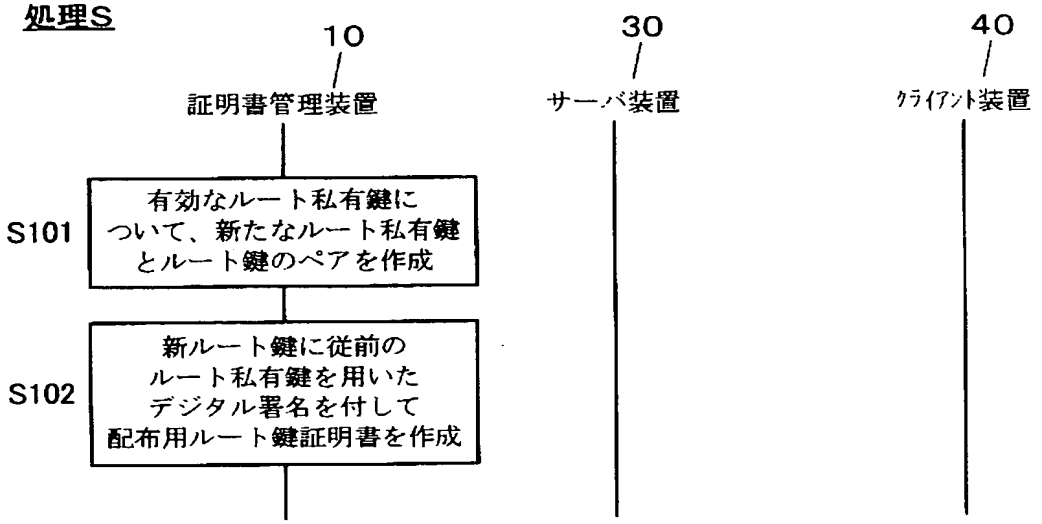
（A）


10
30
40
証明書管理装置
サーバ装置
ｸﾗｲｱﾝﾄ装置
管理装置側要求ａ
応答ａ又は
遅延通知ａ′

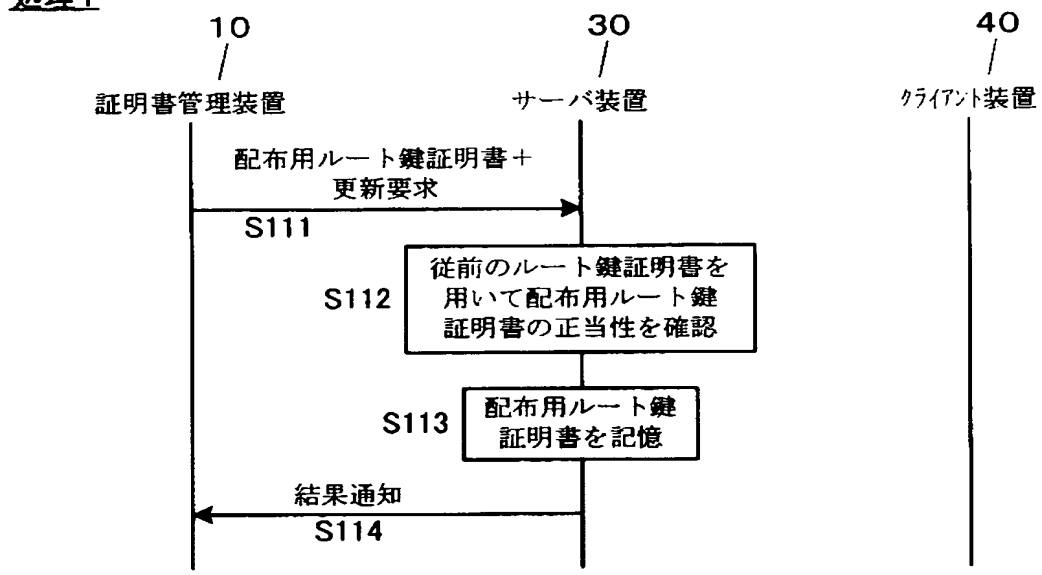
（B）


10
30
40
証明書管理装置
サーバ装置
ｸﾗｲｱﾝﾄ装置
ｸﾗｲｱﾝﾄ装置側要求ｂ
応答ｂ又は
遅延通知ｂ′

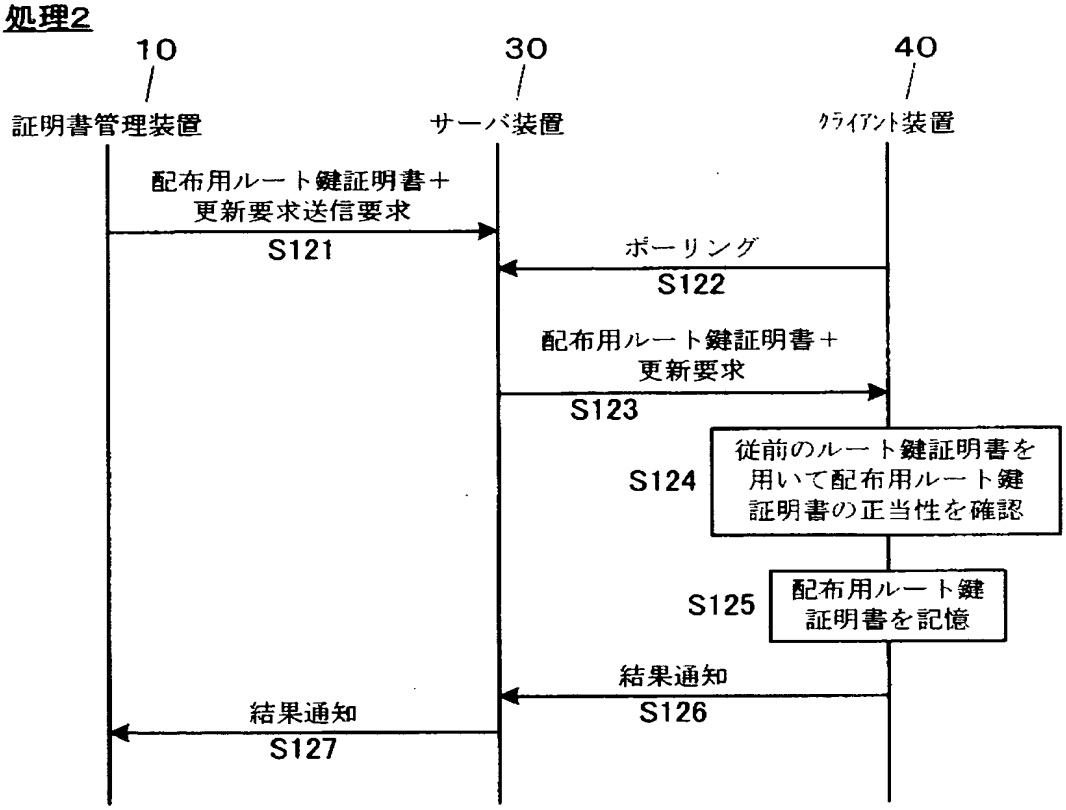
【図４】


処理S
10
証明書管理装置
30
サーバ装置
40
ｸﾗｲｱﾝﾄ装置
S101
有効なルート私有鍵について、新たなルート私有鍵とルート鍵のペアを作成
S102
新ルート鍵に従前のルート私有鍵を用いたデジタル署名を付して配布用ルート鍵証明書を作成

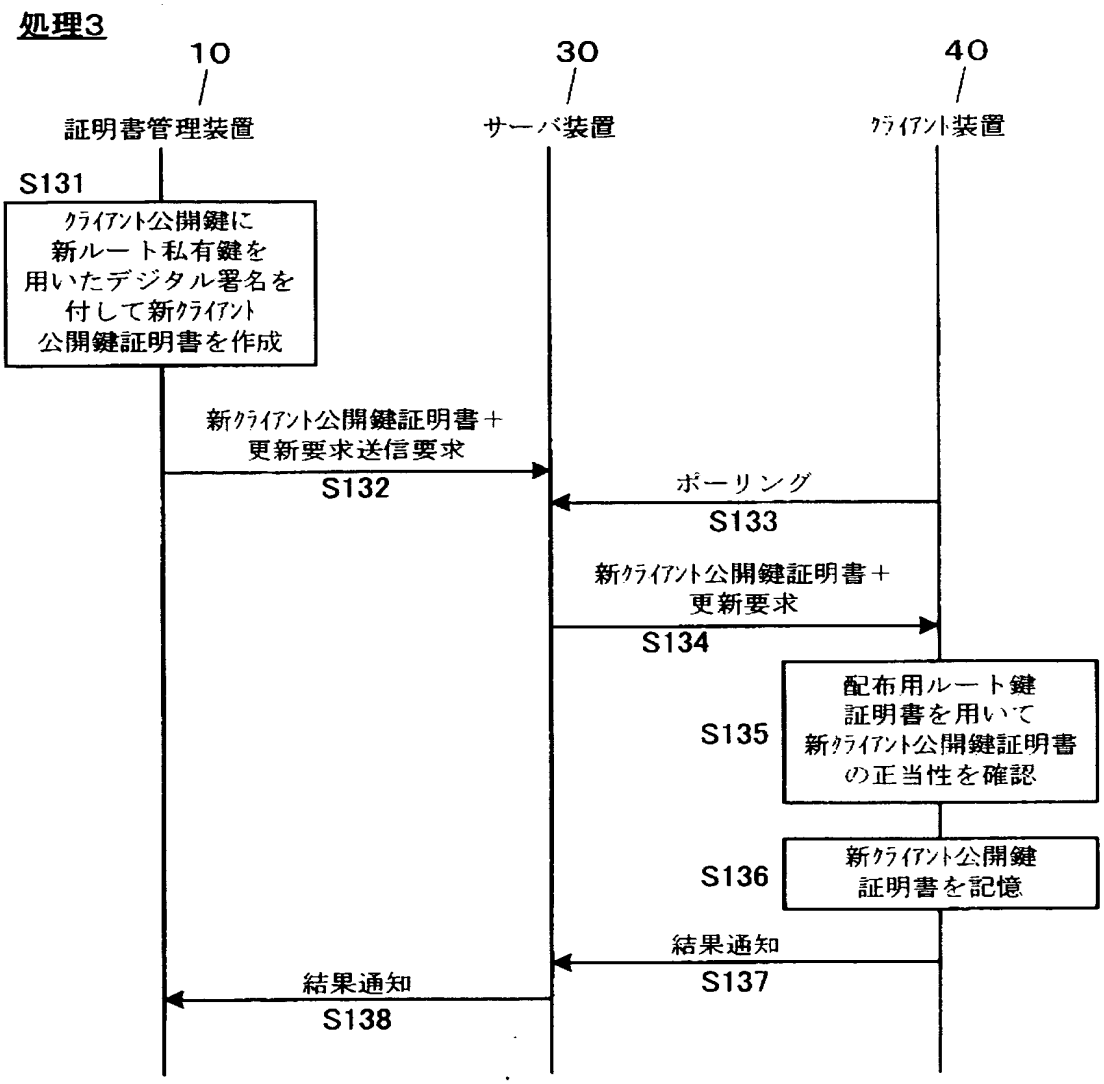
【図５】


処理1
10
証明書管理装置
30
サーバ装置
40
ｸﾗｲｱﾝﾄ装置
配布用ルート鍵証明書＋更新要求
S111
S112
従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認
S113
配布用ルート鍵証明書を記憶
結果通知
S114

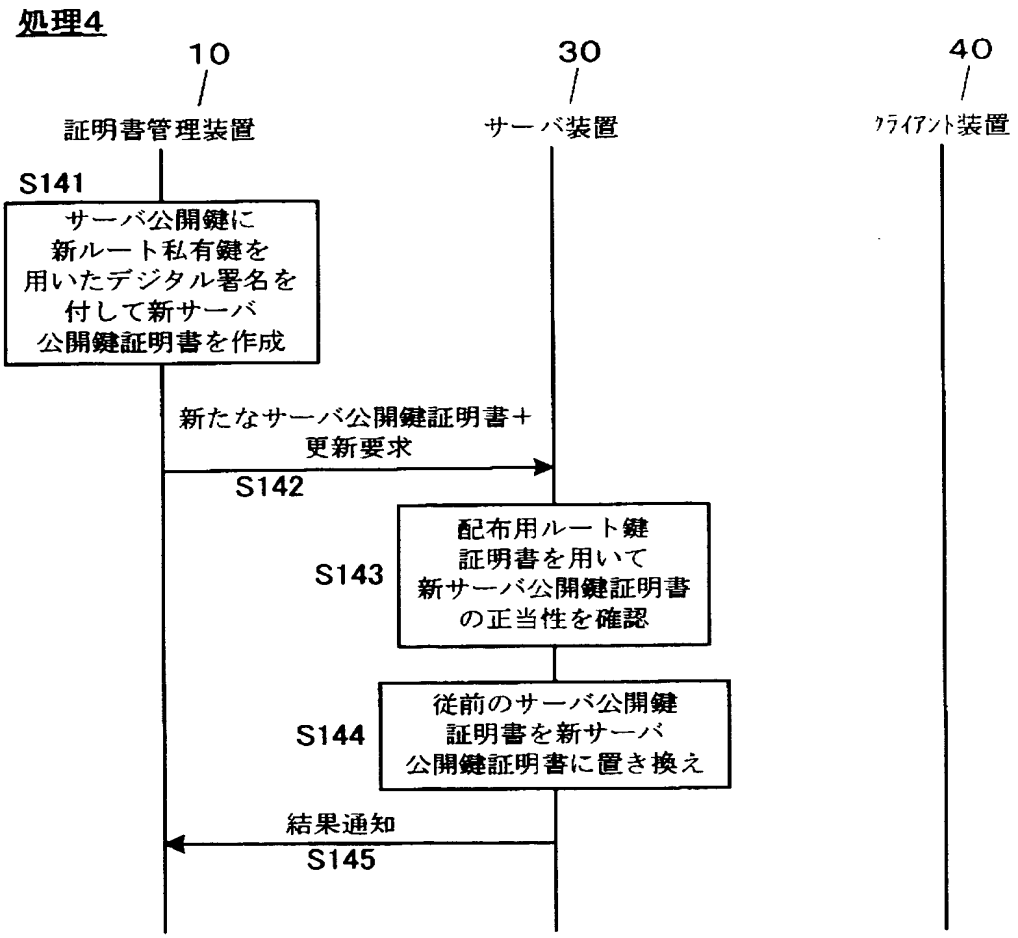
【図６】


処理2
10
30
40
証明書管理装置
サーバ装置
ｸﾗｲｱﾝﾄ装置
配布用ルート鍵証明書＋
更新要求送信要求
S121
ポーリング
S122
配布用ルート鍵証明書＋
更新要求
S123
S124
従前のルート鍵証明書を
用いて配布用ルート鍵
証明書の正当性を確認
S125
配布用ルート鍵
証明書を記憶
結果通知
S126
結果通知
S127

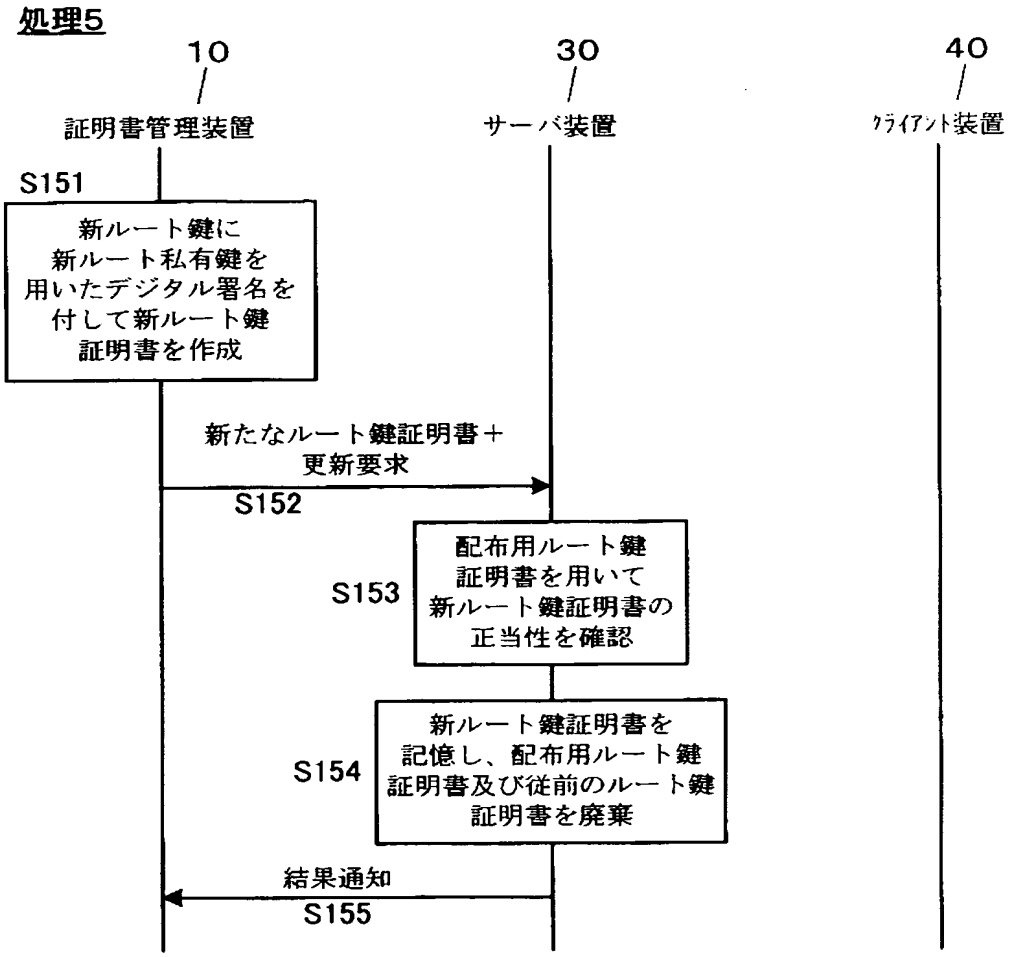
【図７】


処理3
10
30
40
証明書管理装置
サーバ装置
ｸﾗｲｱﾝﾄ装置
S131
ｸﾗｲｱﾝﾄ公開鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新ｸﾗｲｱﾝﾄ
公開鍵証明書を作成
新ｸﾗｲｱﾝﾄ公開鍵証明書＋
更新要求送信要求
S132
ポーリング
S133
新ｸﾗｲｱﾝﾄ公開鍵証明書＋
更新要求
S134
S135
配布用ルート鍵
証明書を用いて
新ｸﾗｲｱﾝﾄ公開鍵証明書
の正当性を確認
S136
新ｸﾗｲｱﾝﾄ公開鍵
証明書を記憶
結果通知
S137
結果通知
S138

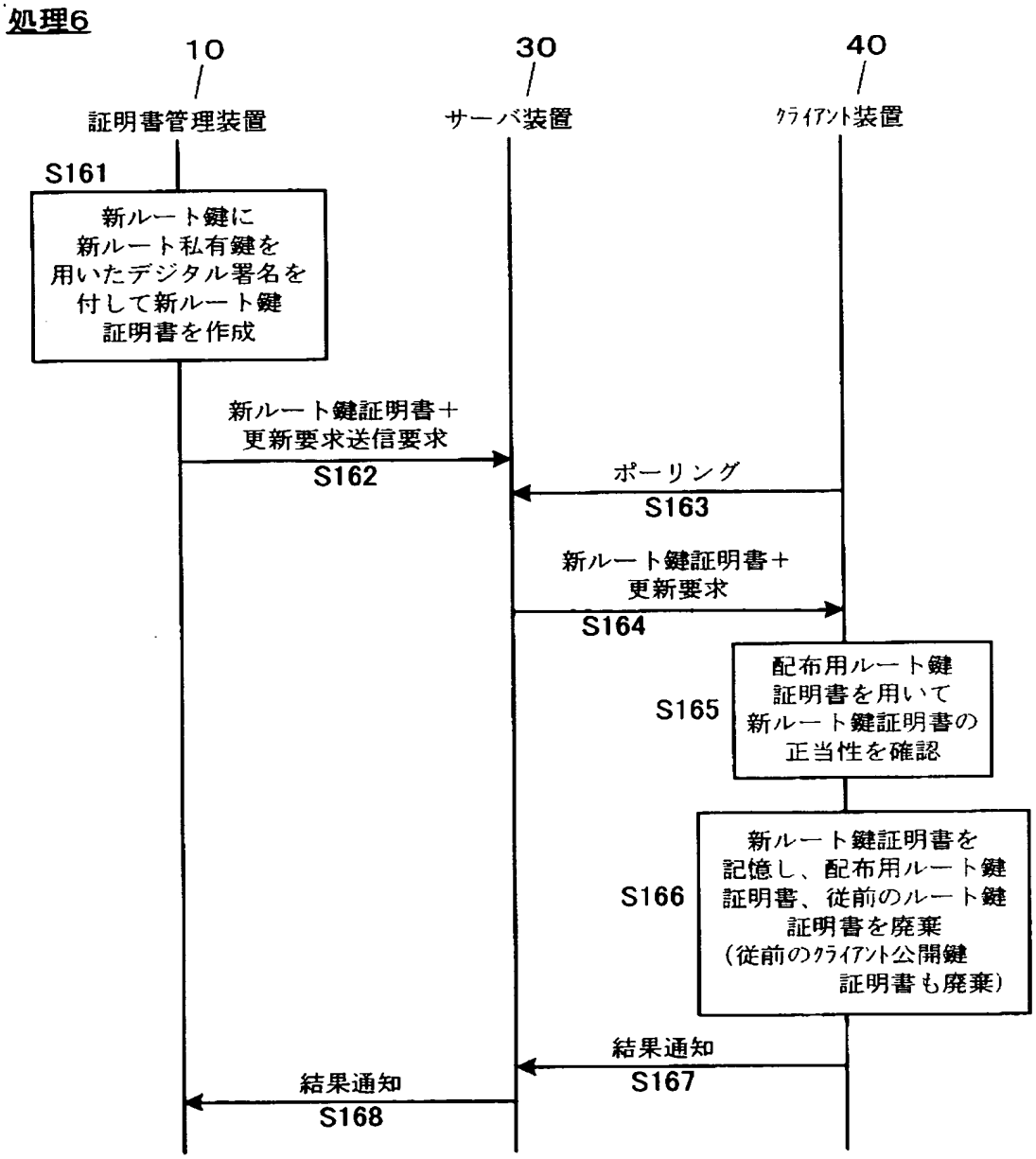
【図８】


処理4
10
30
40
証明書管理装置
サーバ装置
ｸﾗｲｱﾝﾄ装置
S141
サーバ公開鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新サーバ
公開鍵証明書を作成
新たなサーバ公開鍵証明書＋
更新要求
S142
S143
配布用ルート鍵
証明書を用いて
新サーバ公開鍵証明書
の正当性を確認
S144
従前のサーバ公開鍵
証明書を新サーバ
公開鍵証明書に置き換え
結果通知
S145

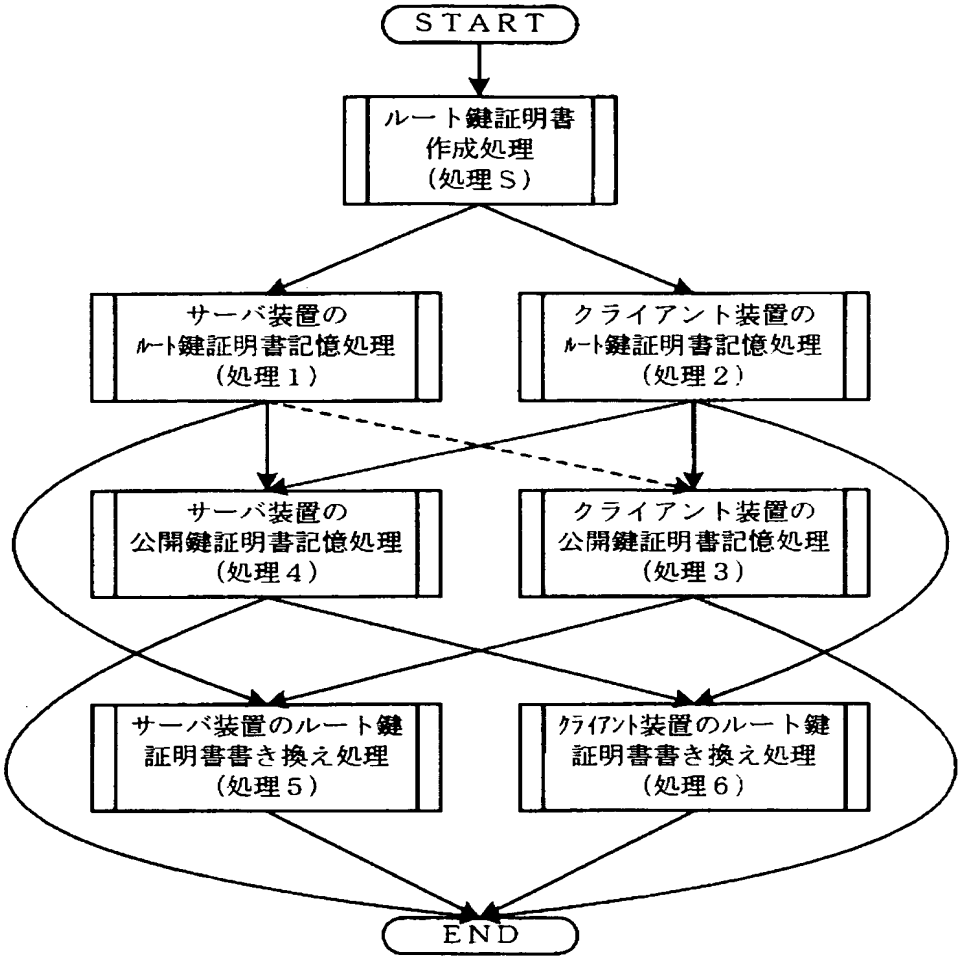
【図９】


処理5
10
30
40
証明書管理装置
サーバ装置
ｸﾗｲｱﾝﾄ装置
S151
新ルート鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新ルート鍵
証明書を作成
新たなルート鍵証明書＋
更新要求
S152
S153
配布用ルート鍵
証明書を用いて
新ルート鍵証明書の
正当性を確認
S154
新ルート鍵証明書を
記憶し、配布用ルート鍵
証明書及び従前のルート鍵
証明書を廃棄
結果通知
S155

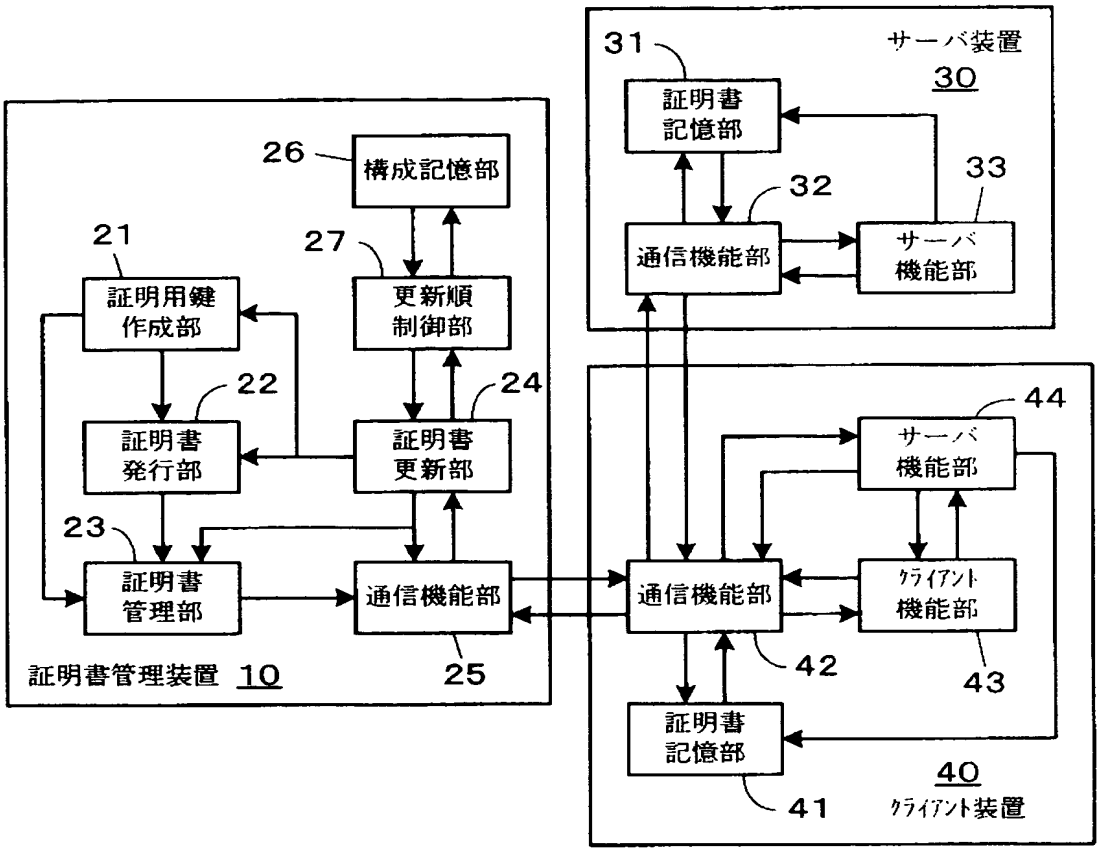
【図１０】


処理6
10
30
40
証明書管理装置
サーバ装置
ｸﾗｲｱﾝﾄ装置
S161
新ルート鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新ルート鍵
証明書を作成
新ルート鍵証明書＋
更新要求送信要求
S162
ポーリング
S163
新ルート鍵証明書＋
更新要求
S164
S165
配布用ルート鍵
証明書を用いて
新ルート鍵証明書の
正当性を確認
S166
新ルート鍵証明書を
記憶し、配布用ルート鍵
証明書、従前のルート鍵
証明書を廃棄
（従前のｸﾗｲｱﾝﾄ公開鍵
証明書も廃棄）
結果通知
S167
結果通知
S168

【図１１】

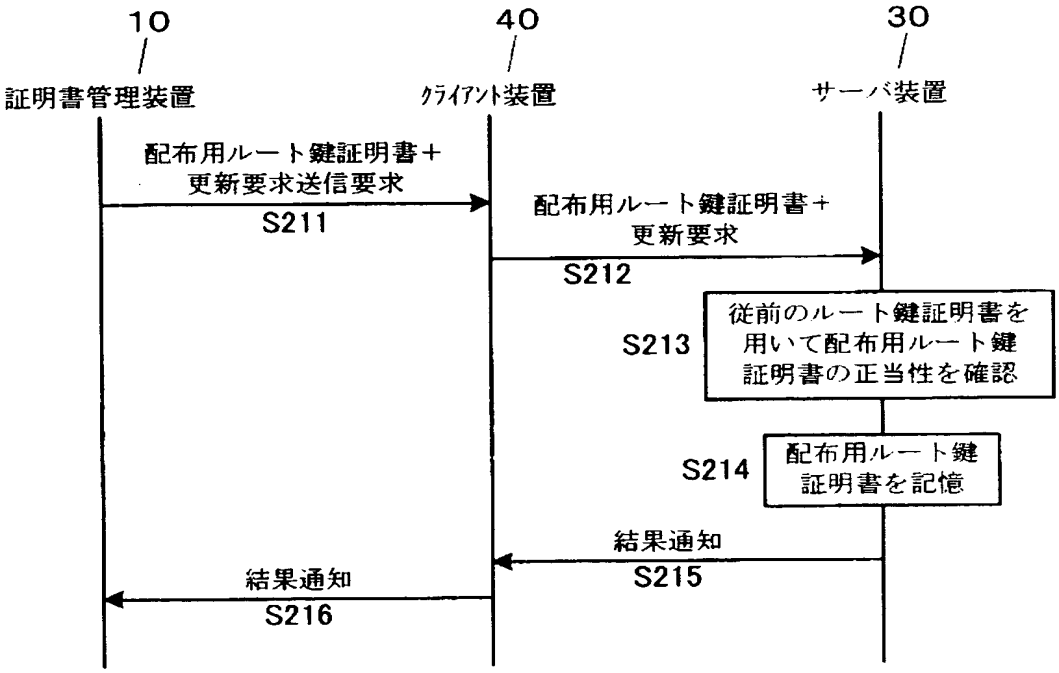

START
ルート鍵証明書
作成処理
（処理Ｓ）
サーバ装置の
ﾙｰﾄ鍵証明書記憶処理
（処理１）
クライアント装置の
ﾙｰﾄ鍵証明書記憶処理
（処理２）
サーバ装置の
公開鍵証明書記憶処理
（処理４）
クライアント装置の
公開鍵証明書記憶処理
（処理３）
サーバ装置のルート鍵
証明書書き換え処理
（処理５）
ｸﾗｲｱﾝﾄ装置のルート鍵
証明書書き換え処理
（処理６）
END

【図１２】


サーバ装置
30
31
証明書
記憶部
32
33
通信機能部
サーバ
機能部
26
構成記憶部
21
27
証明用鍵
作成部
更新順
制御部
22
24
証明書
発行部
証明書
更新部
23
証明書
管理部
通信機能部
25
証明書管理装置 10
通信機能部
44
サーバ
機能部
ｸﾗｲｱﾝﾄ
機能部
42
43
証明書
記憶部
41
40
ｸﾗｲｱﾝﾄ装置

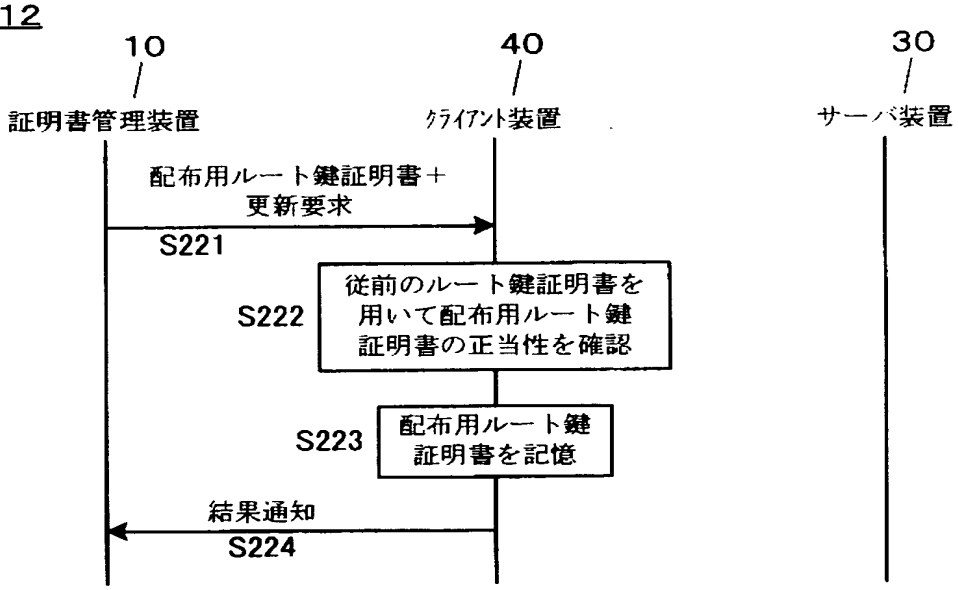
【図１３】

処理11

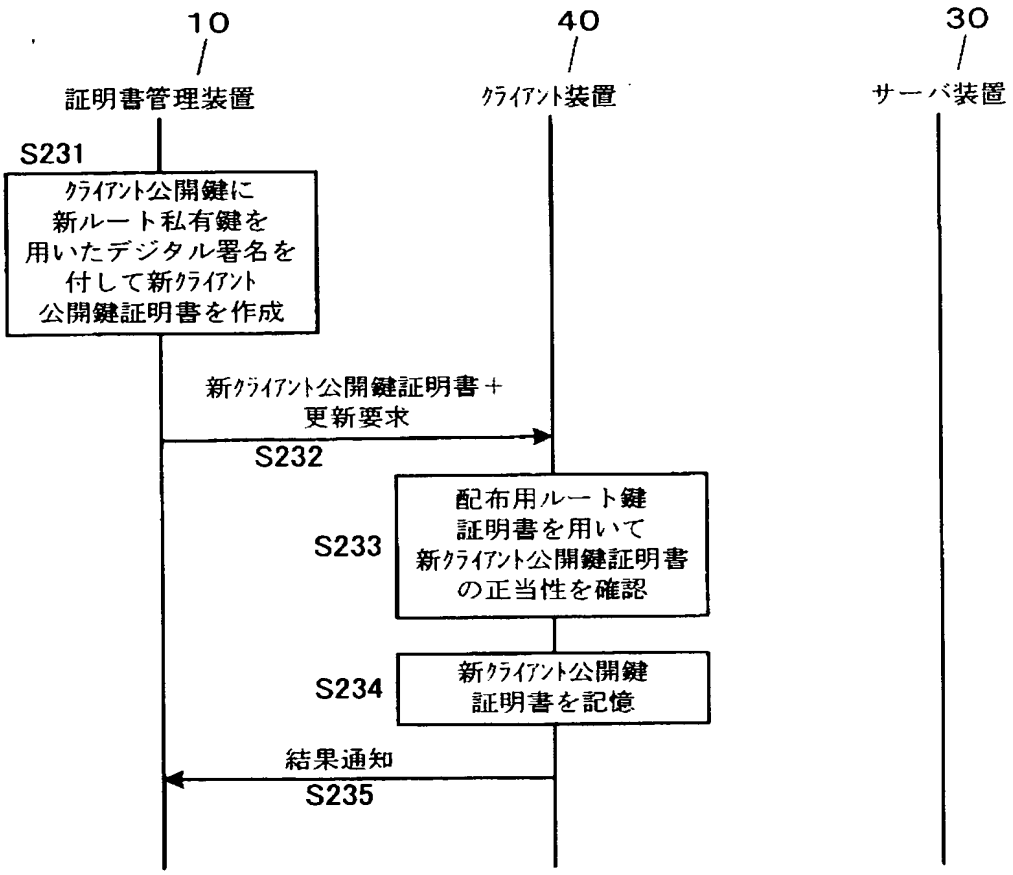


【図１４】

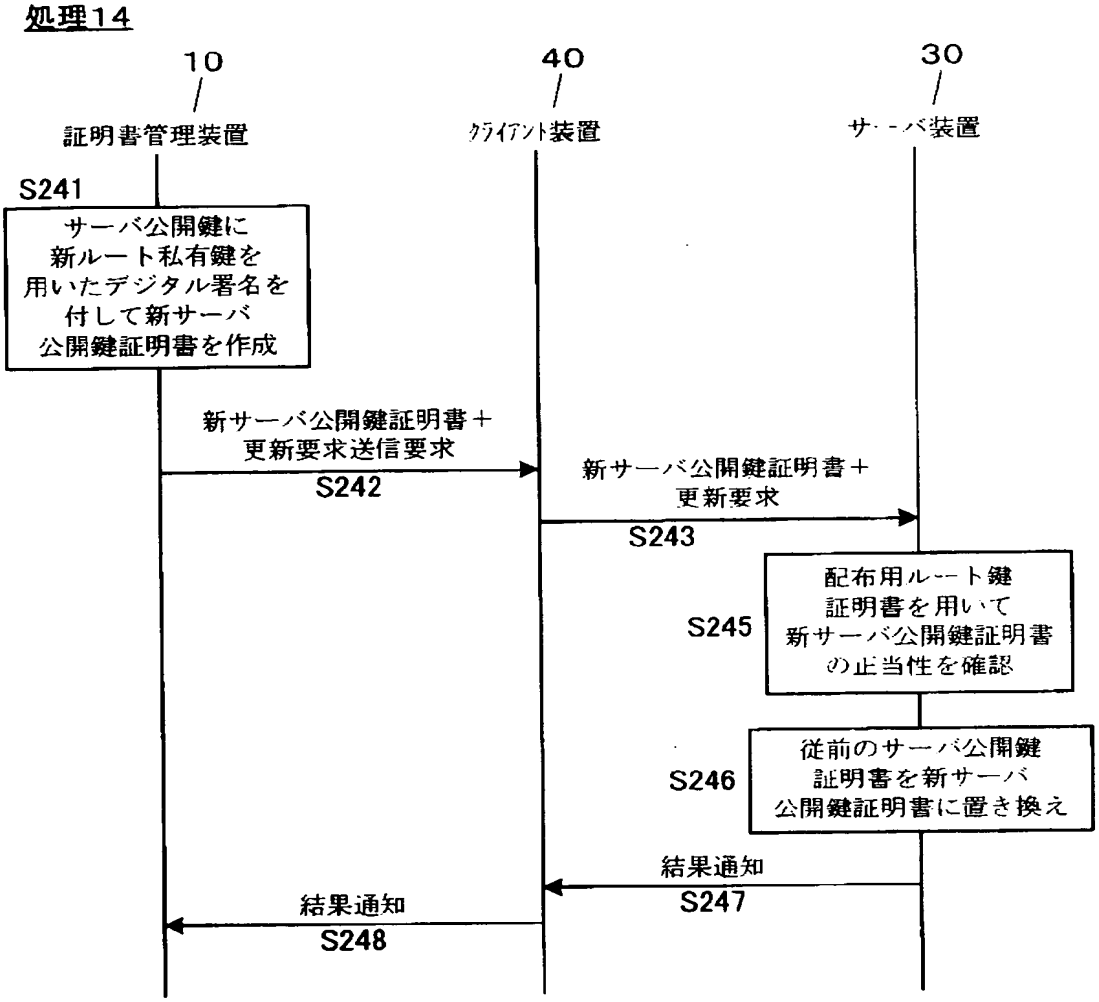
処理12



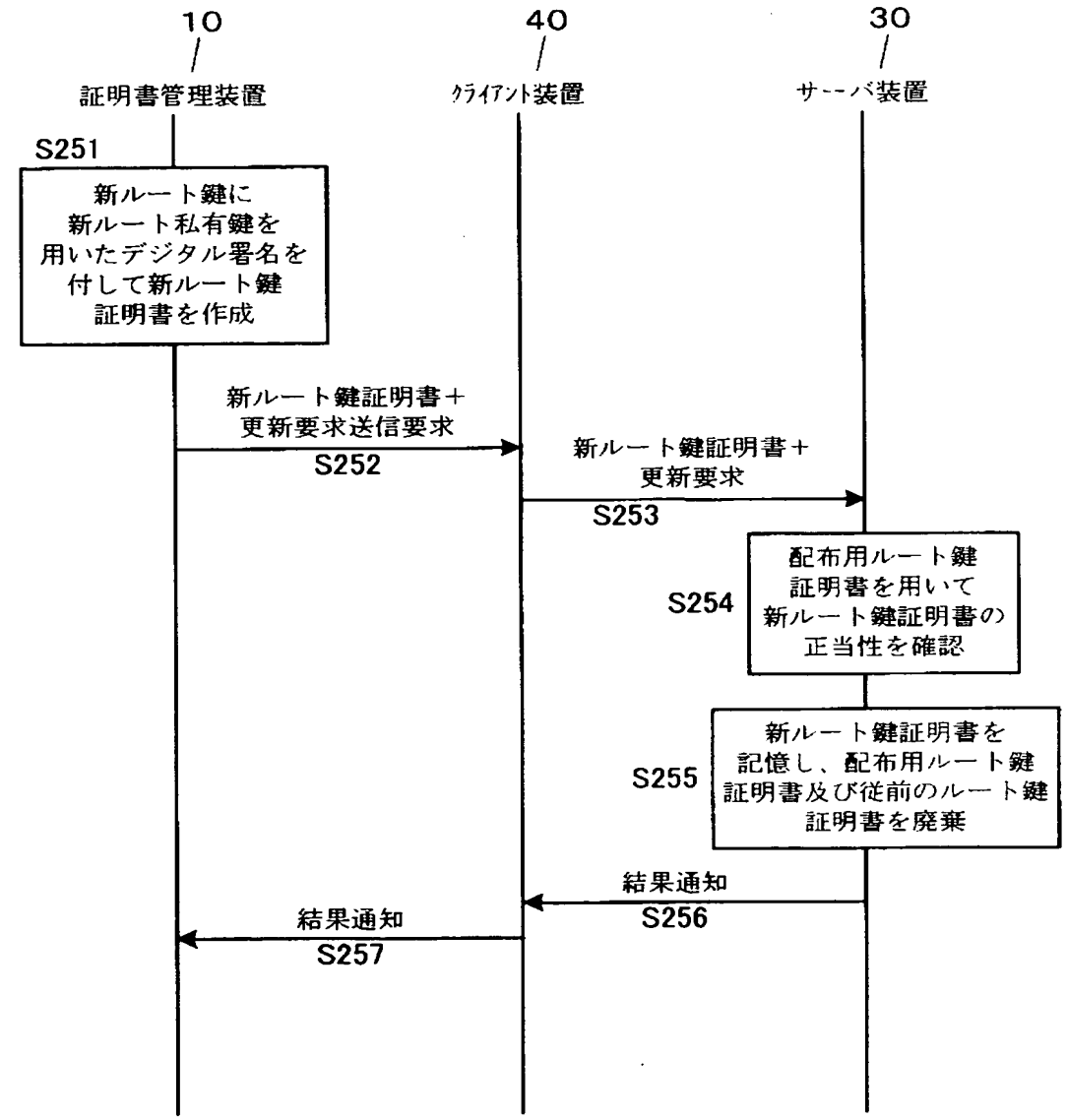
【図１５】


処理13
10
証明書管理装置
40
ｸﾗｲｱﾝﾄ装置
30
サーバ装置
S231
ｸﾗｲｱﾝﾄ公開鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新ｸﾗｲｱﾝﾄ
公開鍵証明書を作成
新ｸﾗｲｱﾝﾄ公開鍵証明書＋
更新要求
S232
S233
配布用ルート鍵
証明書を用いて
新ｸﾗｲｱﾝﾄ公開鍵証明書
の正当性を確認
S234
新ｸﾗｲｱﾝﾄ公開鍵
証明書を記憶
結果通知
S235

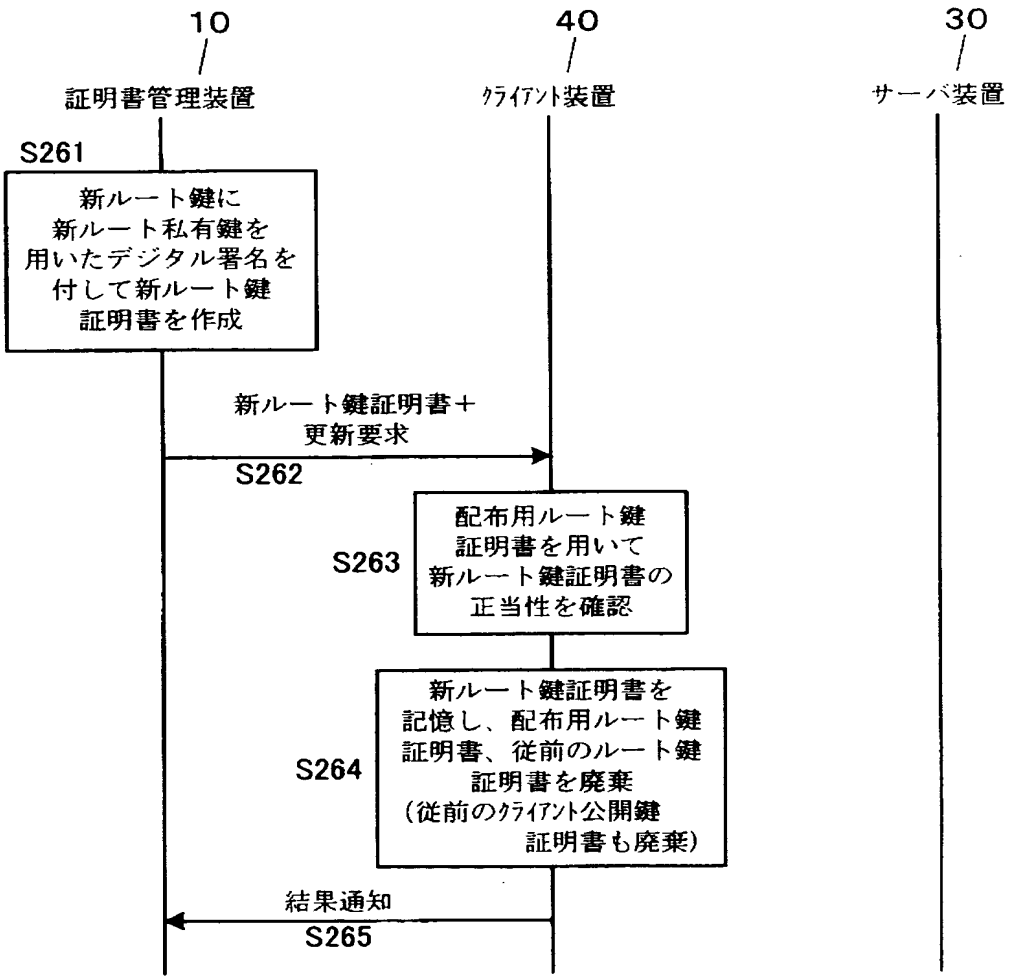
【図１６】


処理14
10
証明書管理装置
40
ｸﾗｲｱﾝﾄ装置
30
ｻｰﾊﾞ装置
S241
サーバ公開鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新サーバ
公開鍵証明書を作成
新サーバ公開鍵証明書＋
更新要求送信要求
S242
新サーバ公開鍵証明書＋
更新要求
S243
S245
配布用ルート鍵
証明書を用いて
新サーバ公開鍵証明書
の正当性を確認
S246
従前のサーバ公開鍵
証明書を新サーバ
公開鍵証明書に置き換え
結果通知
S247
結果通知
S248

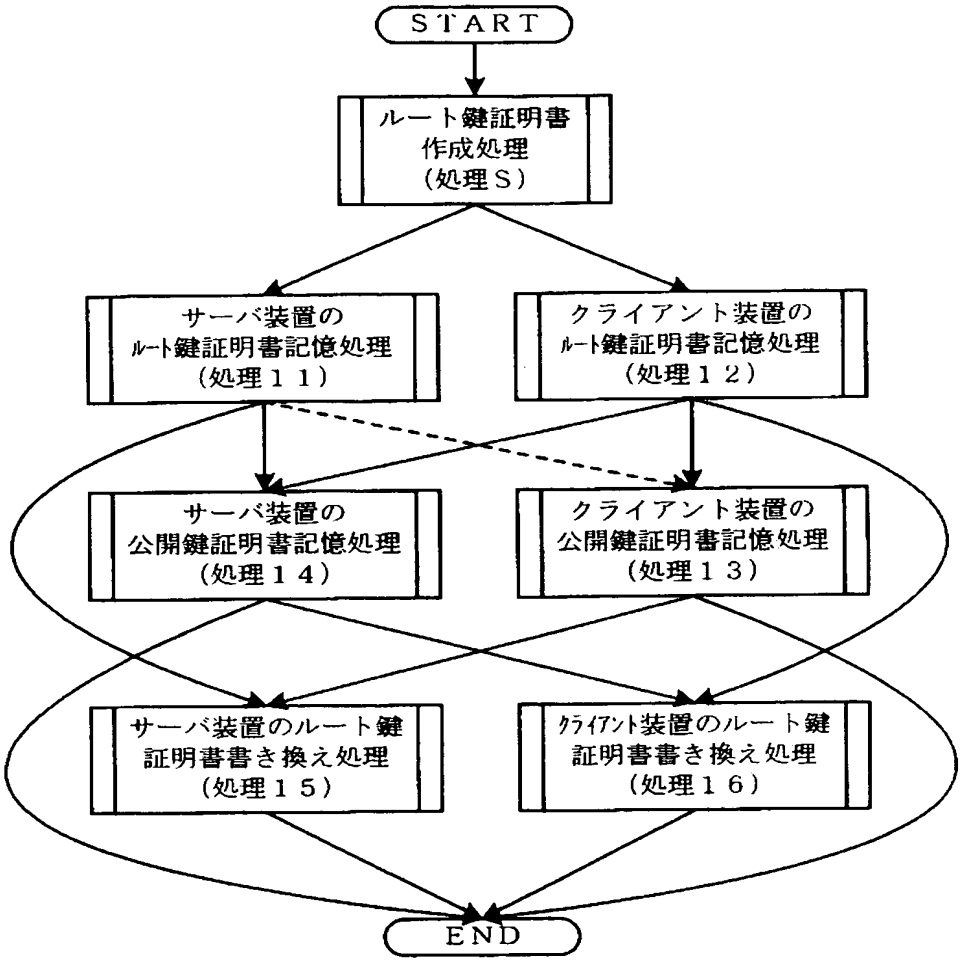
【図１７】


処理15
10
40
30
証明書管理装置
ｸﾗｲｱﾝﾄ装置
ｻｰﾊﾞ装置
S251
新ルート鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新ルート鍵
証明書を作成
新ルート鍵証明書＋
更新要求送信要求
S252
新ルート鍵証明書＋
更新要求
S253
S254
配布用ルート鍵
証明書を用いて
新ルート鍵証明書の
正当性を確認
S255
新ルート鍵証明書を
記憶し、配布用ルート鍵
証明書及び従前のルート鍵
証明書を廃棄
結果通知
S256
結果通知
S257

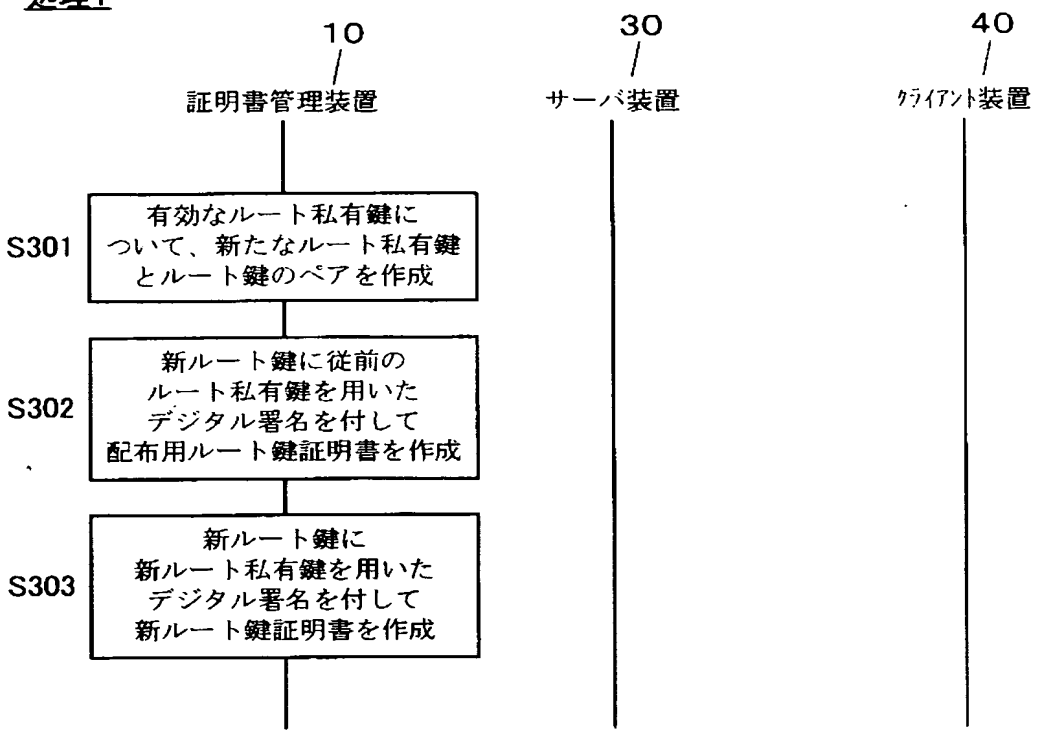
【図１８】


処理16
10
40
30
証明書管理装置
ｸﾗｲｱﾝﾄ装置
サーバ装置
S261
新ルート鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新ルート鍵
証明書を作成
新ルート鍵証明書＋
更新要求
S262
S263
配布用ルート鍵
証明書を用いて
新ルート鍵証明書の
正当性を確認
S264
新ルート鍵証明書を
記憶し、配布用ルート鍵
証明書、従前のルート鍵
証明書を廃棄
（従前のｸﾗｲｱﾝﾄ公開鍵
証明書も廃棄）
結果通知
S265

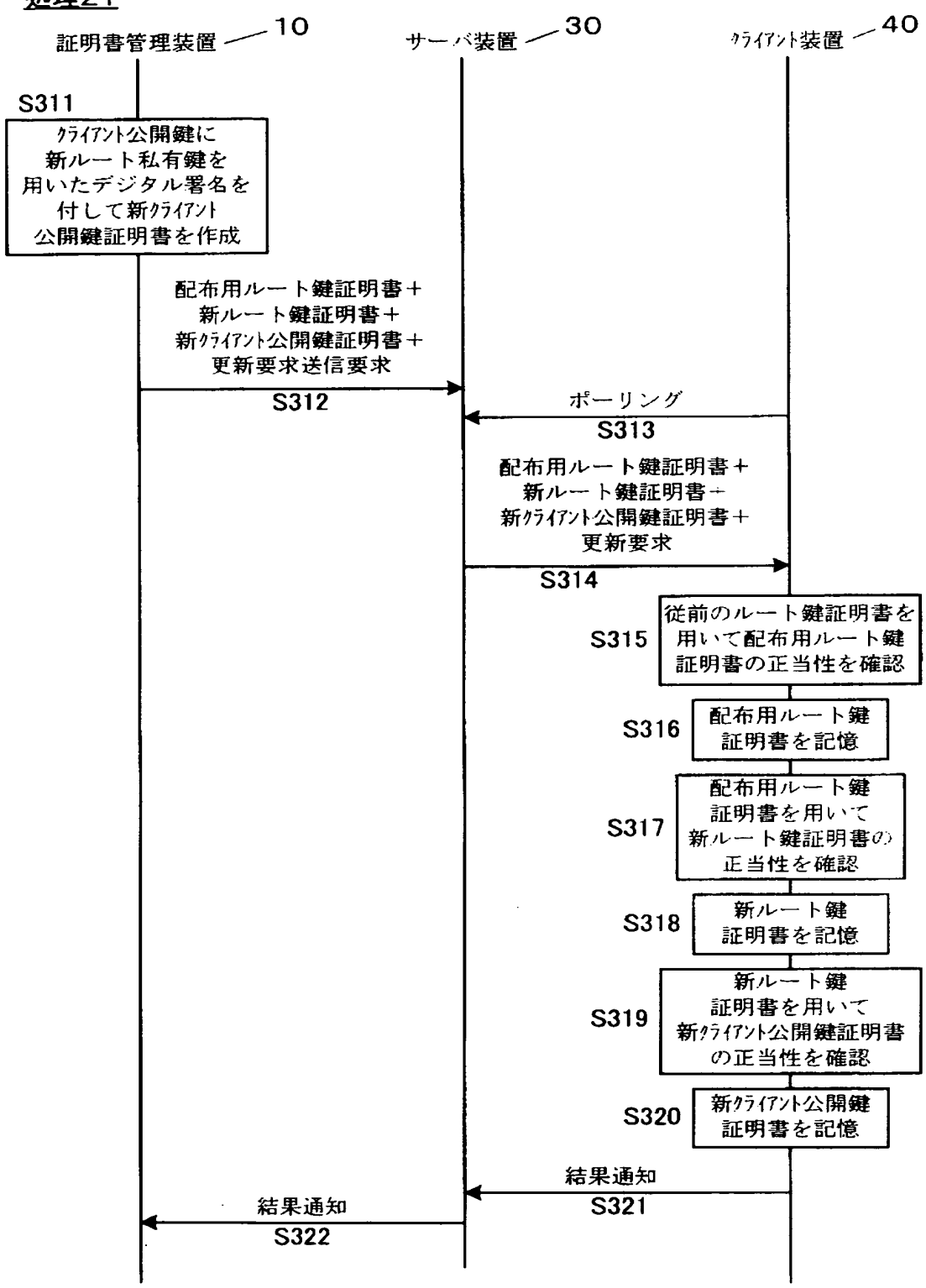
【図１９】


START
ルート鍵証明書
作成処理
（処理Ｓ）
サーバ装置の
ルート鍵証明書記憶処理
（処理１１）
クライアント装置の
ルート鍵証明書記憶処理
（処理１２）
サーバ装置の
公開鍵証明書記憶処理
（処理１４）
クライアント装置の
公開鍵証明書記憶処理
（処理１３）
サーバ装置のルート鍵
証明書書き換え処理
（処理１５）
ｸﾗｲｱﾝﾄ装置のルート鍵
証明書書き換え処理
（処理１６）
END

【図２０】

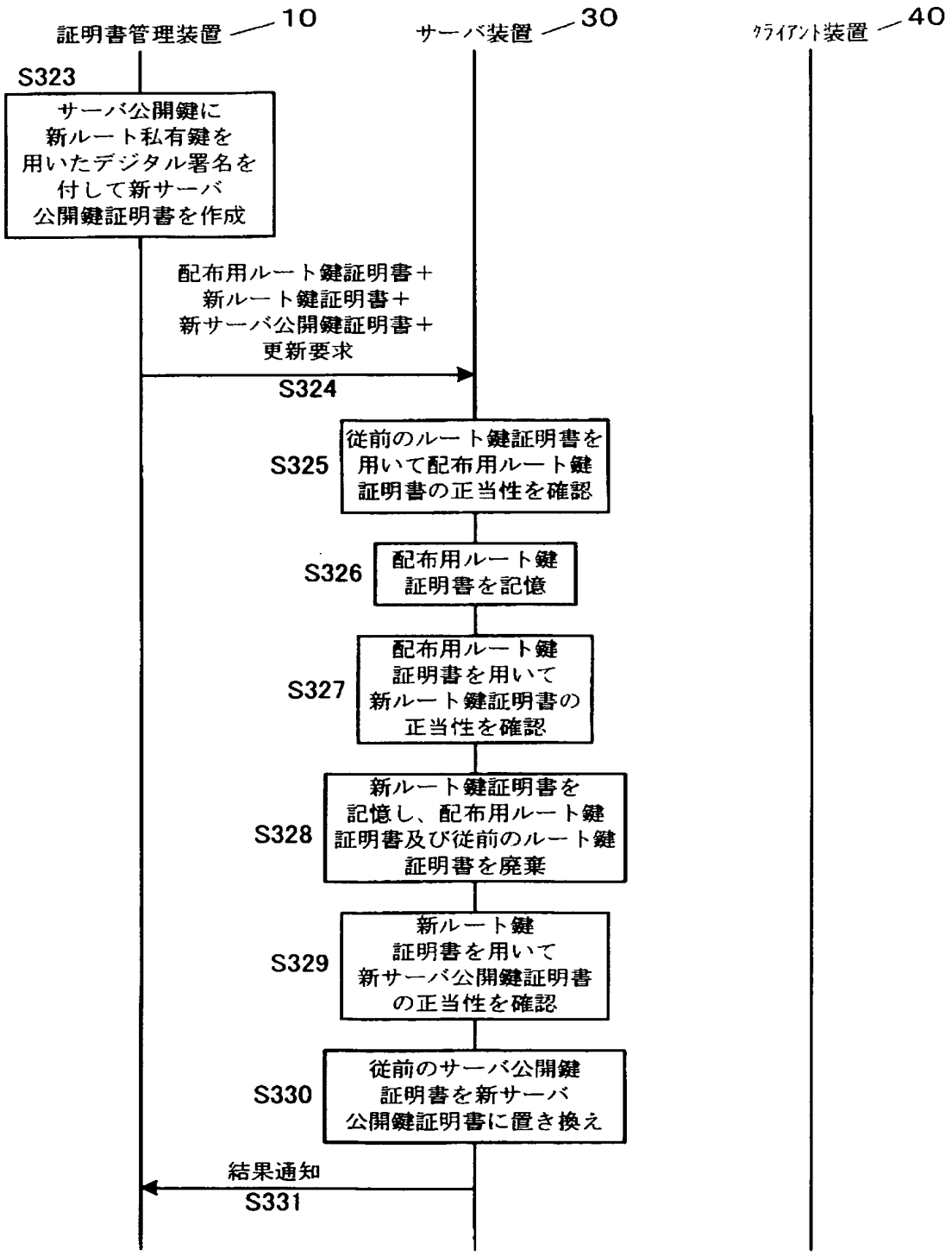

処理T
10
証明書管理装置
30
サーバ装置
40
ｸﾗｲｱﾝﾄ装置
S301
有効なルート私有鍵について、新たなルート私有鍵とルート鍵のペアを作成
S302
新ルート鍵に従前のルート私有鍵を用いたデジタル署名を付して配布用ルート鍵証明書を作成
S303
新ルート鍵に新ルート私有鍵を用いたデジタル署名を付して新ルート鍵証明書を作成

【図２１】


処理21
証明書管理装置
10
サーバ装置
30
ｸﾗｲｱﾝﾄ装置
40
S311
ｸﾗｲｱﾝﾄ公開鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新ｸﾗｲｱﾝﾄ
公開鍵証明書を作成
配布用ルート鍵証明書＋
新ルート鍵証明書＋
新ｸﾗｲｱﾝﾄ公開鍵証明書＋
更新要求送信要求
S312
ポーリング
S313
配布用ルート鍵証明書＋
新ルート鍵証明書＋
新ｸﾗｲｱﾝﾄ公開鍵証明書＋
更新要求
S314
S315
従前のルート鍵証明書を
用いて配布用ルート鍵
証明書の正当性を確認
S316
配布用ルート鍵
証明書を記憶
S317
配布用ルート鍵
証明書を用いて
新ルート鍵証明書の
正当性を確認
S318
新ルート鍵
証明書を記憶
S319
新ルート鍵
証明書を用いて
新ｸﾗｲｱﾝﾄ公開鍵証明書
の正当性を確認
S320
新ｸﾗｲｱﾝﾄ公開鍵
証明書を記憶
結果通知
S321
結果通知
S322

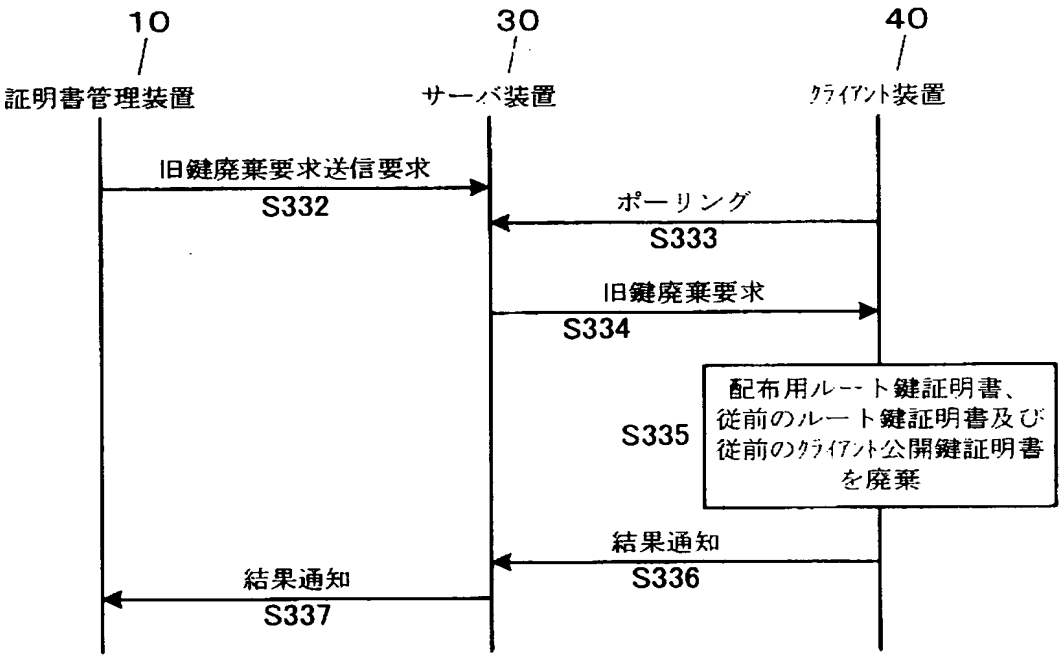
【図２２】

処理22


証明書管理装置
10
サーバ装置
30
ｸﾗｲｱﾝﾄ装置
40
S323
サーバ公開鍵に新ルート私有鍵を用いたデジタル署名を付して新サーバ公開鍵証明書を作成
配布用ルート鍵証明書＋新ルート鍵証明書＋新サーバ公開鍵証明書＋更新要求
S324
S325
従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認
S326
配布用ルート鍵証明書を記憶
S327
配布用ルート鍵証明書を用いて新ルート鍵証明書の正当性を確認
S328
新ルート鍵証明書を記憶し、配布用ルート鍵証明書及び従前のルート鍵証明書を廃棄
S329
新ルート鍵証明書を用いて新サーバ公開鍵証明書の正当性を確認
S330
従前のサーバ公開鍵証明書を新サーバ公開鍵証明書に置き換え
結果通知
S331

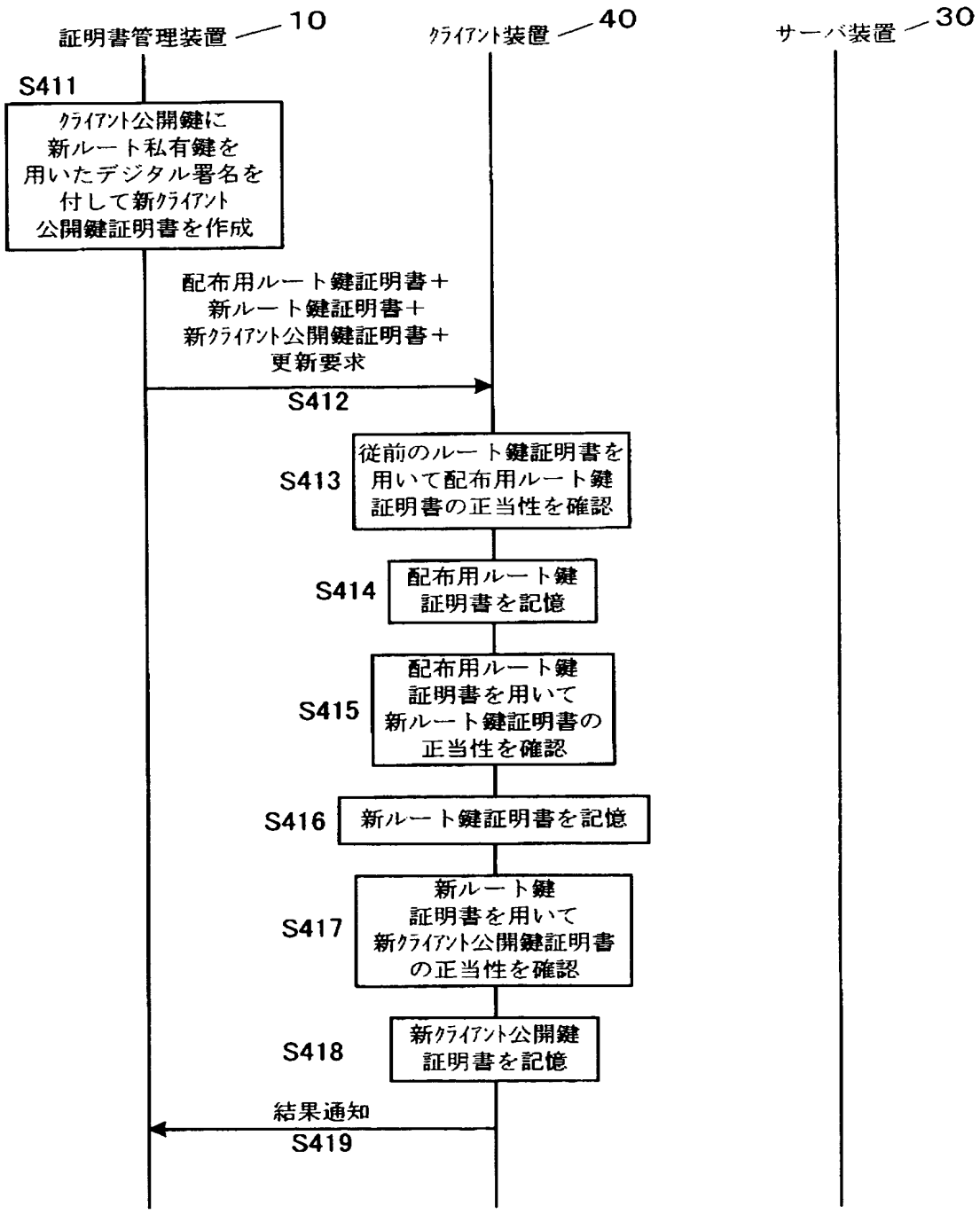
【図２３】

処理23

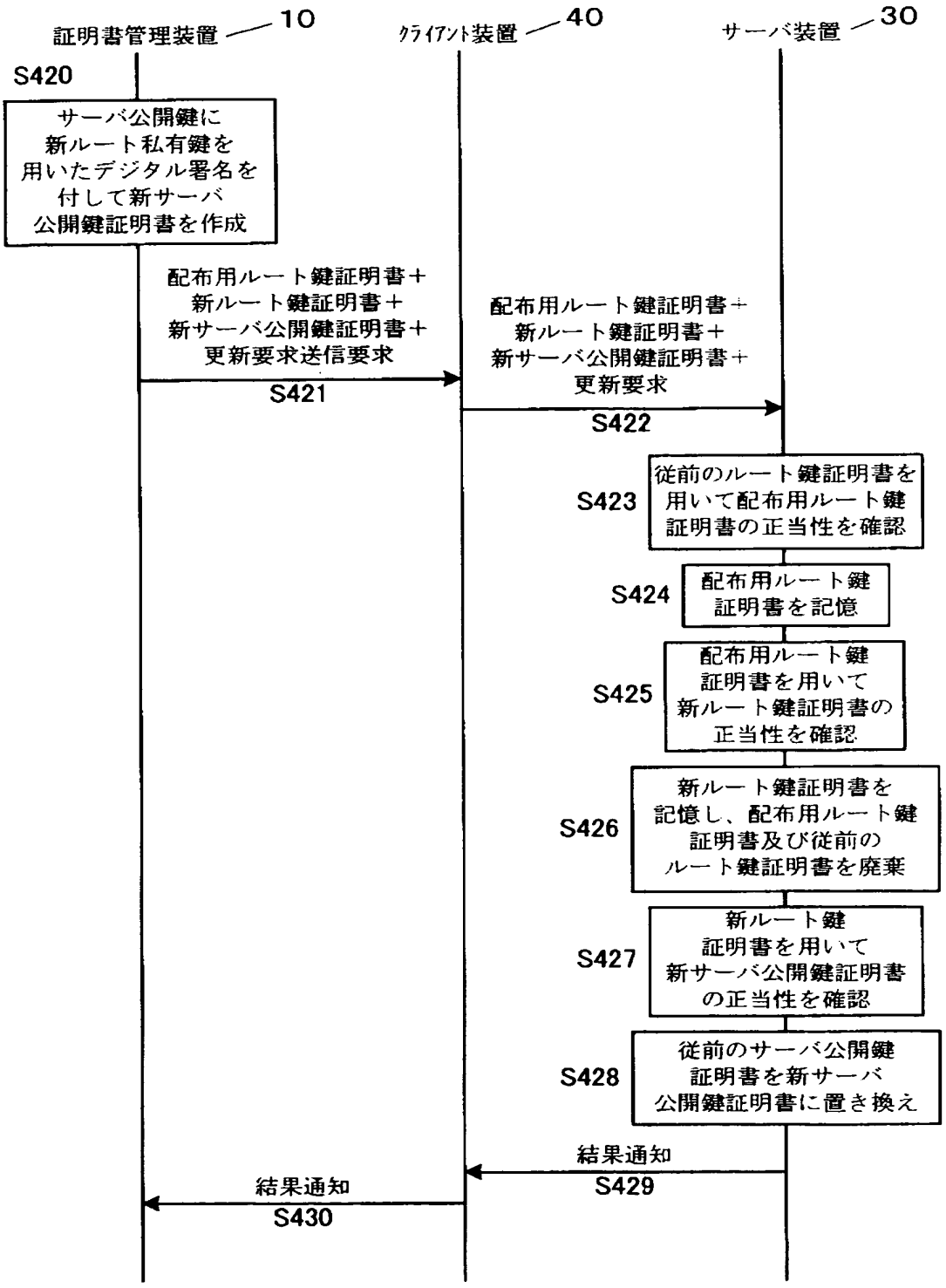

10
30
40
証明書管理装置
サーバ装置
ｸﾗｲｱﾝﾄ装置
旧鍵廃棄要求送信要求
S332
ポーリング
S333
旧鍵廃棄要求
S334
S335
配布用ルート鍵証明書、
従前のルート鍵証明書及び
従前のｸﾗｲｱﾝﾄ公開鍵証明書
を廃棄
結果通知
S336
結果通知
S337

【図２４】

処理31

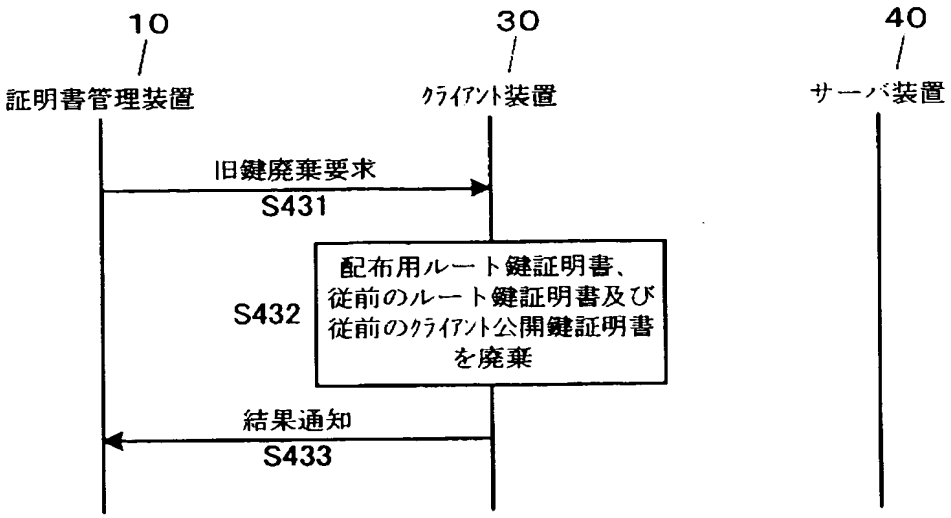

証明書管理装置
10
ｸﾗｲｱﾝﾄ装置
40
サーバ装置
30
S411
ｸﾗｲｱﾝﾄ公開鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新ｸﾗｲｱﾝﾄ
公開鍵証明書を作成
配布用ルート鍵証明書＋
新ルート鍵証明書＋
新ｸﾗｲｱﾝﾄ公開鍵証明書＋
更新要求
S412
S413
従前のルート鍵証明書を
用いて配布用ルート鍵
証明書の正当性を確認
S414
配布用ルート鍵
証明書を記憶
S415
配布用ルート鍵
証明書を用いて
新ルート鍵証明書の
正当性を確認
S416
新ルート鍵証明書を記憶
S417
新ルート鍵
証明書を用いて
新ｸﾗｲｱﾝﾄ公開鍵証明書
の正当性を確認
S418
新ｸﾗｲｱﾝﾄ公開鍵
証明書を記憶
結果通知
S419

【図２５】

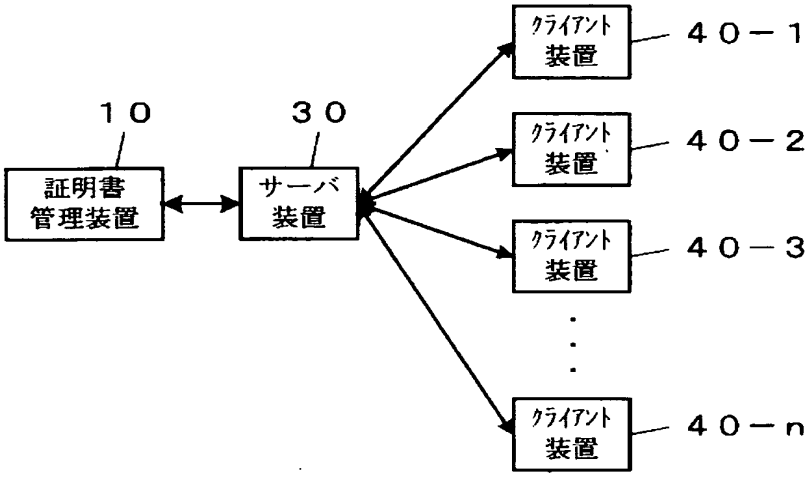

処理32
証明書管理装置
10
クライアント装置
40
サーバ装置
30
S420
サーバ公開鍵に新ルート私有鍵を用いたデジタル署名を付して新サーバ公開鍵証明書を作成
配布用ルート鍵証明書＋新ルート鍵証明書＋新サーバ公開鍵証明書＋更新要求送信要求
S421
配布用ルート鍵証明書＋新ルート鍵証明書＋新サーバ公開鍵証明書＋更新要求
S422
S423
従前のルート鍵証明書を用いて配布用ルート鍵証明書の正当性を確認
S424
配布用ルート鍵証明書を記憶
S425
配布用ルート鍵証明書を用いて新ルート鍵証明書の正当性を確認
S426
新ルート鍵証明書を記憶し、配布用ルート鍵証明書及び従前のルート鍵証明書を廃棄
S427
新ルート鍵証明書を用いて新サーバ公開鍵証明書の正当性を確認
S428
従前のサーバ公開鍵証明書を新サーバ公開鍵証明書に置き換え
結果通知
S429
結果通知
S430

【図２６】

処理33


10
30
40
証明書管理装置
ｸﾗｲｱﾝﾄ装置
サーバ装置
旧鍵廃棄要求
S431
S432
配布用ルート鍵証明書、
従前のルート鍵証明書及び
従前のｸﾗｲｱﾝﾄ公開鍵証明書
を廃棄
結果通知
S433

【図２７】



【図２８】

ノードＩＤ	
管理装置との直接通信可否	
通信相手ノードＩＤ	ｸﾗｲｱﾝﾄ/ｻｰﾊﾞ
	使用ルート鍵
	更新状態
・・・	・・・
	・・・
	・・・
・・・	・・・
	・・・
	・・・

【図２９】

（ａ）

サーバ装置３０	
直接通信可	
ｸﾗｲｱﾝﾄ装置４０－１	サーバ
	ルート鍵Ａ
	更新要
ｸﾗｲｱﾝﾄ装置４０－２	サーバ
	ルート鍵Ａ
	更新要
ｸﾗｲｱﾝﾄ装置４０－３	サーバ
	ルート鍵Ａ
	更新要
・・・	・・・
	・・・
	・・・
ｸﾗｲｱﾝﾄ装置４０－ｎ	サーバ
	ルート鍵Ａ
	更新要

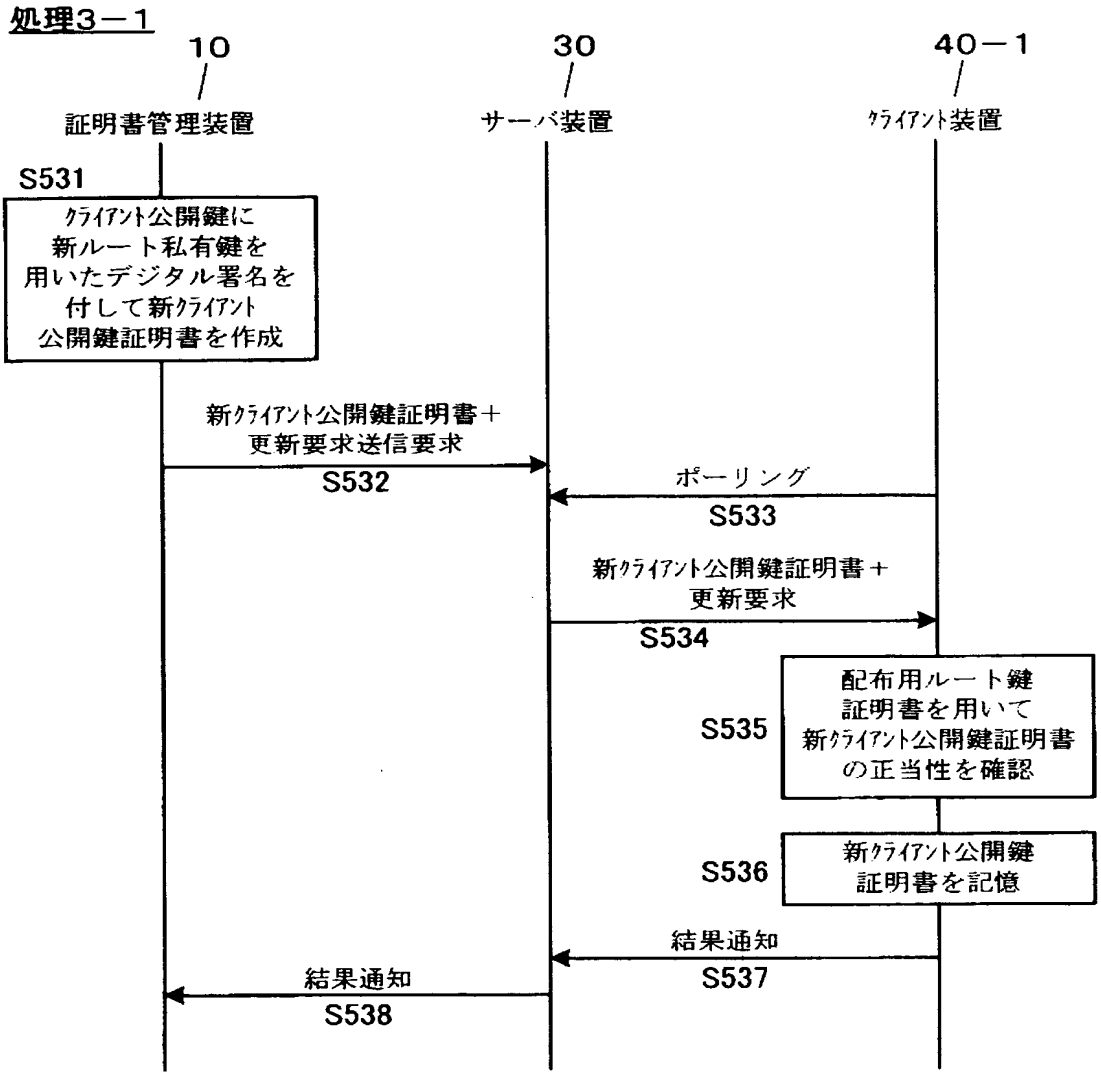
（ｂ）

ｸﾗｲｱﾝﾄ装置４０－１	
直接通信否	
なし	

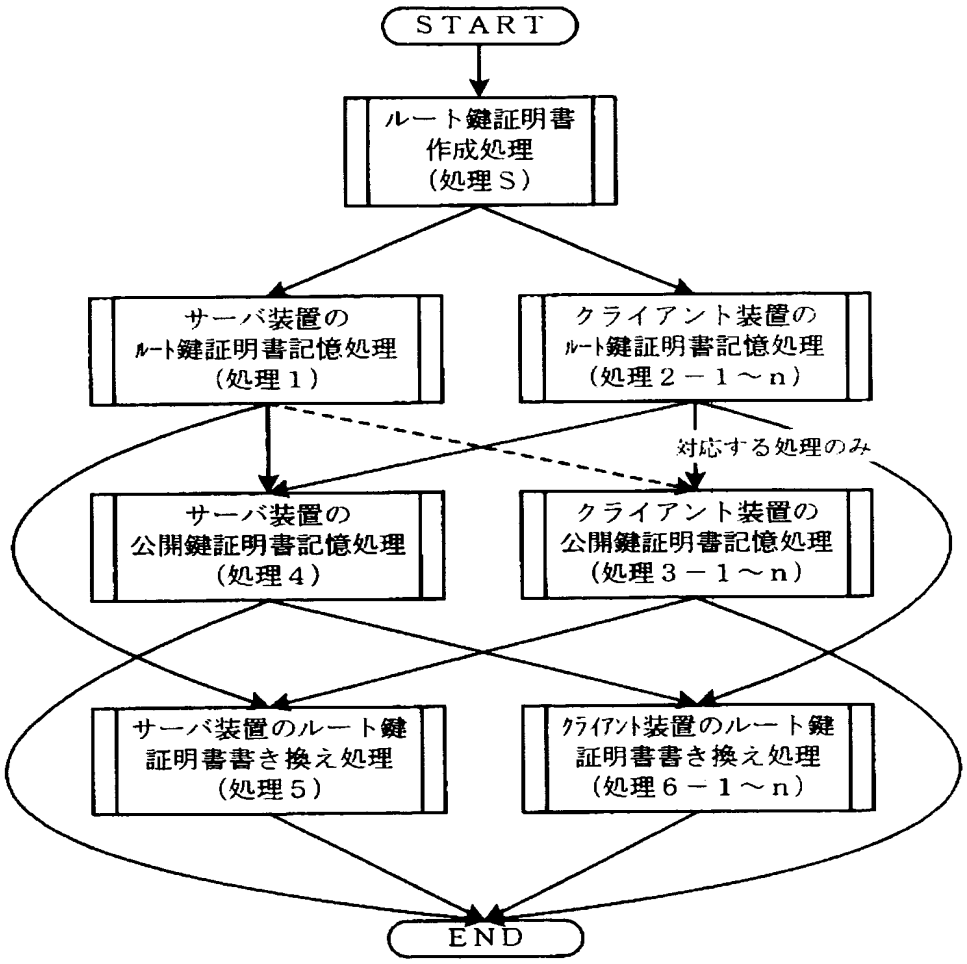
（ｃ）

ｸﾗｲｱﾝﾄ装置４０－１	
直接通信否	
サーバ装置３０	クライアント
	ルート鍵Ａ
	更新要

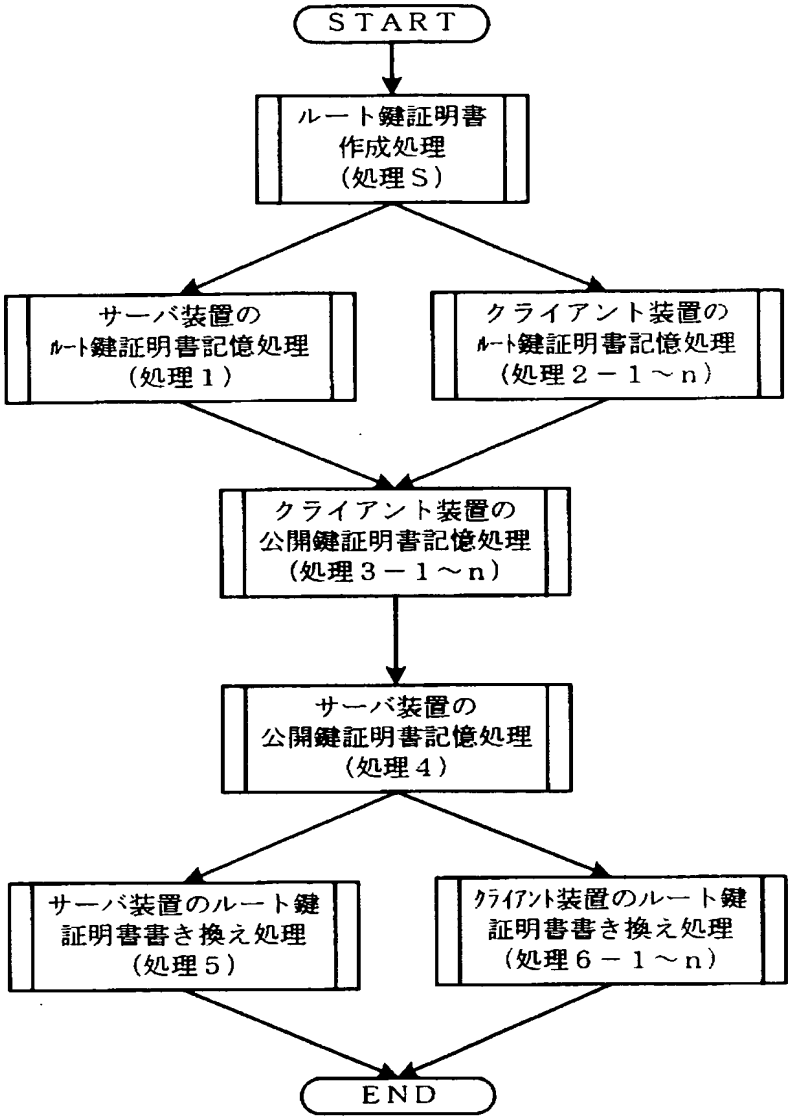
【図３０】


処理3－1
10
証明書管理装置
30
サーバ装置
40－1
ｸﾗｲｱﾝﾄ装置
S531
ｸﾗｲｱﾝﾄ公開鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新ｸﾗｲｱﾝﾄ
公開鍵証明書を作成
新ｸﾗｲｱﾝﾄ公開鍵証明書＋
更新要求送信要求
S532
ポーリング
S533
新ｸﾗｲｱﾝﾄ公開鍵証明書＋
更新要求
S534
S535
配布用ルート鍵
証明書を用いて
新ｸﾗｲｱﾝﾄ公開鍵証明書
の正当性を確認
S536
新ｸﾗｲｱﾝﾄ公開鍵
証明書を記憶
結果通知
S537
結果通知
S538

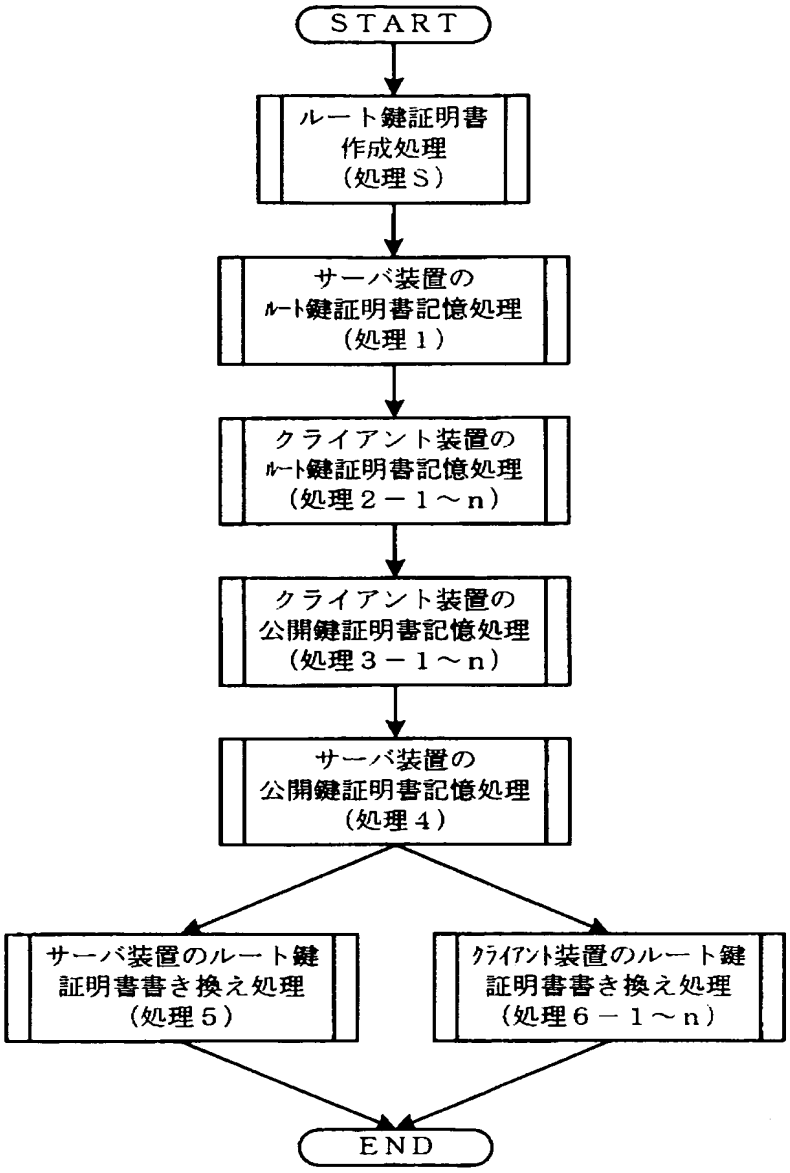
【図３１】


START
ルート鍵証明書
作成処理
（処理Ｓ）
サーバ装置の
ルート鍵証明書記憶処理
（処理１）
クライアント装置の
ルート鍵証明書記憶処理
（処理２－１～ｎ）
対応する処理のみ
サーバ装置の
公開鍵証明書記憶処理
（処理４）
クライアント装置の
公開鍵証明書記憶処理
（処理３－１～ｎ）
サーバ装置のルート鍵
証明書書き換え処理
（処理５）
ｸﾗｲｱﾝﾄ装置のルート鍵
証明書書き換え処理
（処理６－１～ｎ）
END

【図３２】

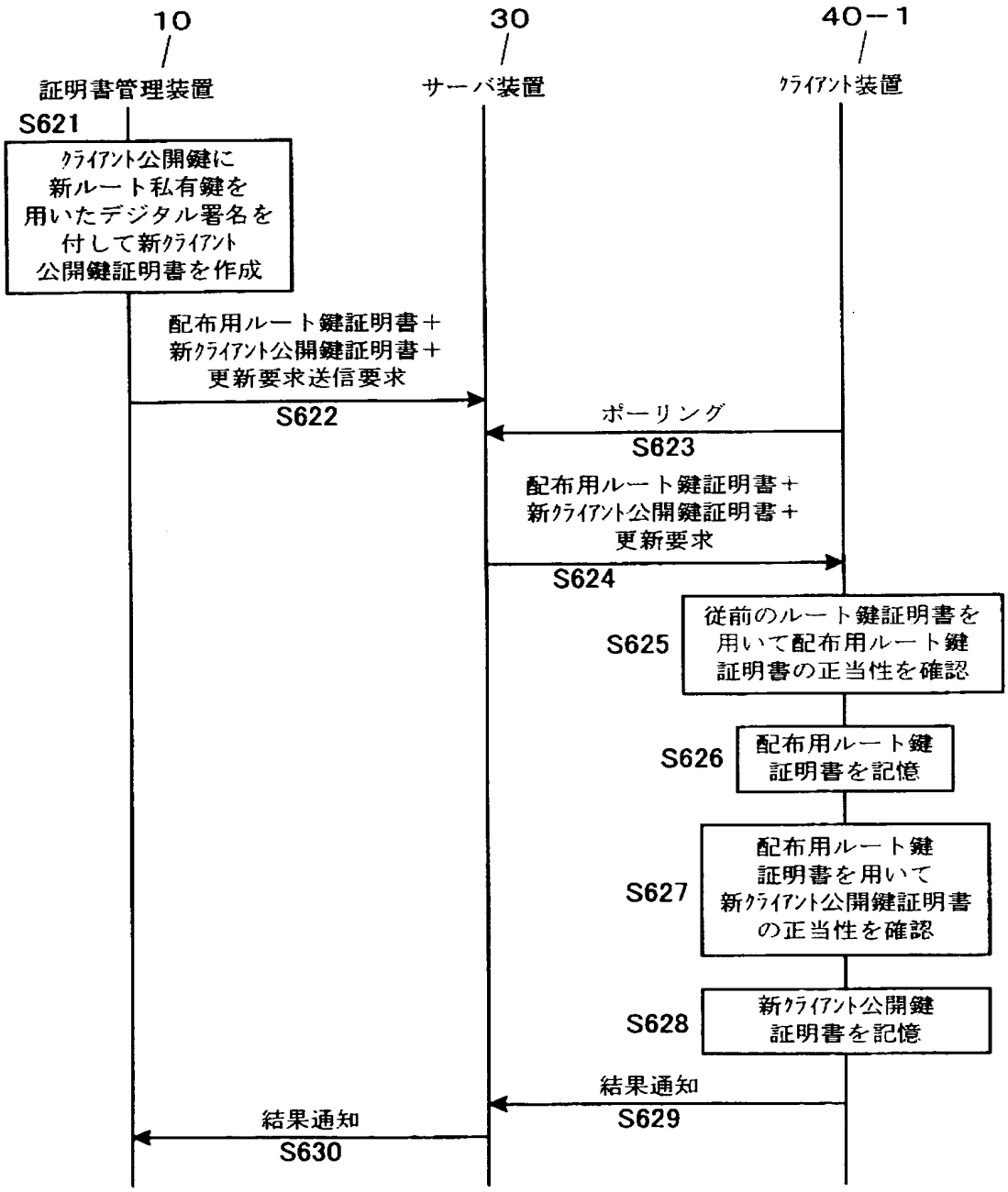

START
ルート鍵証明書
作成処理
（処理Ｓ）
サーバ装置の
ルート鍵証明書記憶処理
（処理１）
クライアント装置の
ルート鍵証明書記憶処理
（処理２－１～ｎ）
クライアント装置の
公開鍵証明書記憶処理
（処理３－１～ｎ）
サーバ装置の
公開鍵証明書記憶処理
（処理４）
サーバ装置のルート鍵
証明書書き換え処理
（処理５）
クライアント装置のルート鍵
証明書書き換え処理
（処理６－１～ｎ）
END

【図３３】

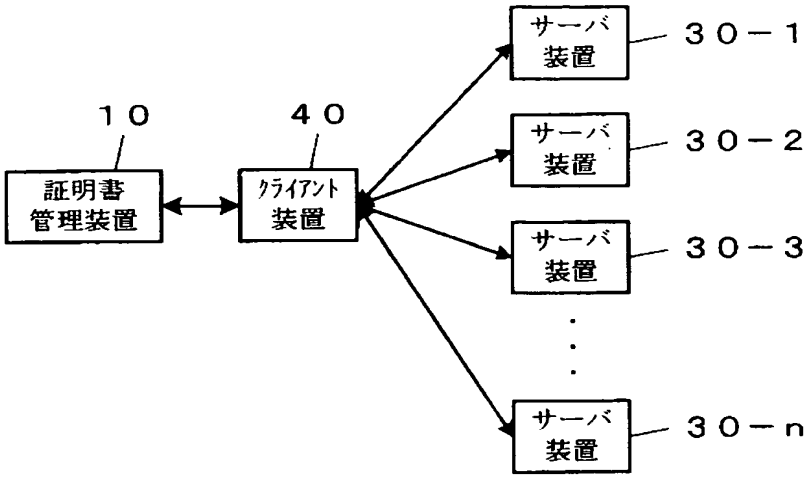

START
ルート鍵証明書
作成処理
（処理Ｓ）
サーバ装置の
ﾙｰﾄ鍵証明書記憶処理
（処理１）
クライアント装置の
ﾙｰﾄ鍵証明書記憶処理
（処理２－１～ｎ）
クライアント装置の
公開鍵証明書記憶処理
（処理３－１～ｎ）
サーバ装置の
公開鍵証明書記憶処理
（処理４）
サーバ装置のルート鍵
証明書書き換え処理
（処理５）
ｸﾗｲｱﾝﾄ装置のルート鍵
証明書書き換え処理
（処理６－１～ｎ）
END

【図３４】

処理2′－1


10
30
40－1
証明書管理装置
サーバ装置
ｸﾗｲｱﾝﾄ装置
S621
ｸﾗｲｱﾝﾄ公開鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新ｸﾗｲｱﾝﾄ
公開鍵証明書を作成
配布用ルート鍵証明書＋
新ｸﾗｲｱﾝﾄ公開鍵証明書＋
更新要求送信要求
S622
ポーリング
S623
配布用ルート鍵証明書＋
新ｸﾗｲｱﾝﾄ公開鍵証明書＋
更新要求
S624
S625
従前のルート鍵証明書を
用いて配布用ルート鍵
証明書の正当性を確認
S626
配布用ルート鍵
証明書を記憶
S627
配布用ルート鍵
証明書を用いて
新ｸﾗｲｱﾝﾄ公開鍵証明書
の正当性を確認
S628
新ｸﾗｲｱﾝﾄ公開鍵
証明書を記憶
結果通知
S629
結果通知
S630

【図３５】



【図３６】

（ａ）

ｸﾗｲｱﾝﾄ装置４０	
直接通信可	
サーバ装置３０－１	クライアント
	ルート鍵Ａ
	更新要
サーバ装置３０－２	クライアント
	ルート鍵Ａ
	更新要
サーバ装置３０－３	クライアント
	ルート鍵Ａ
	更新要
・ ・ ・	・・・
	・・・
	・・・
サーバ装置３０－ｎ	クライアント
	ルート鍵Ａ
	更新要

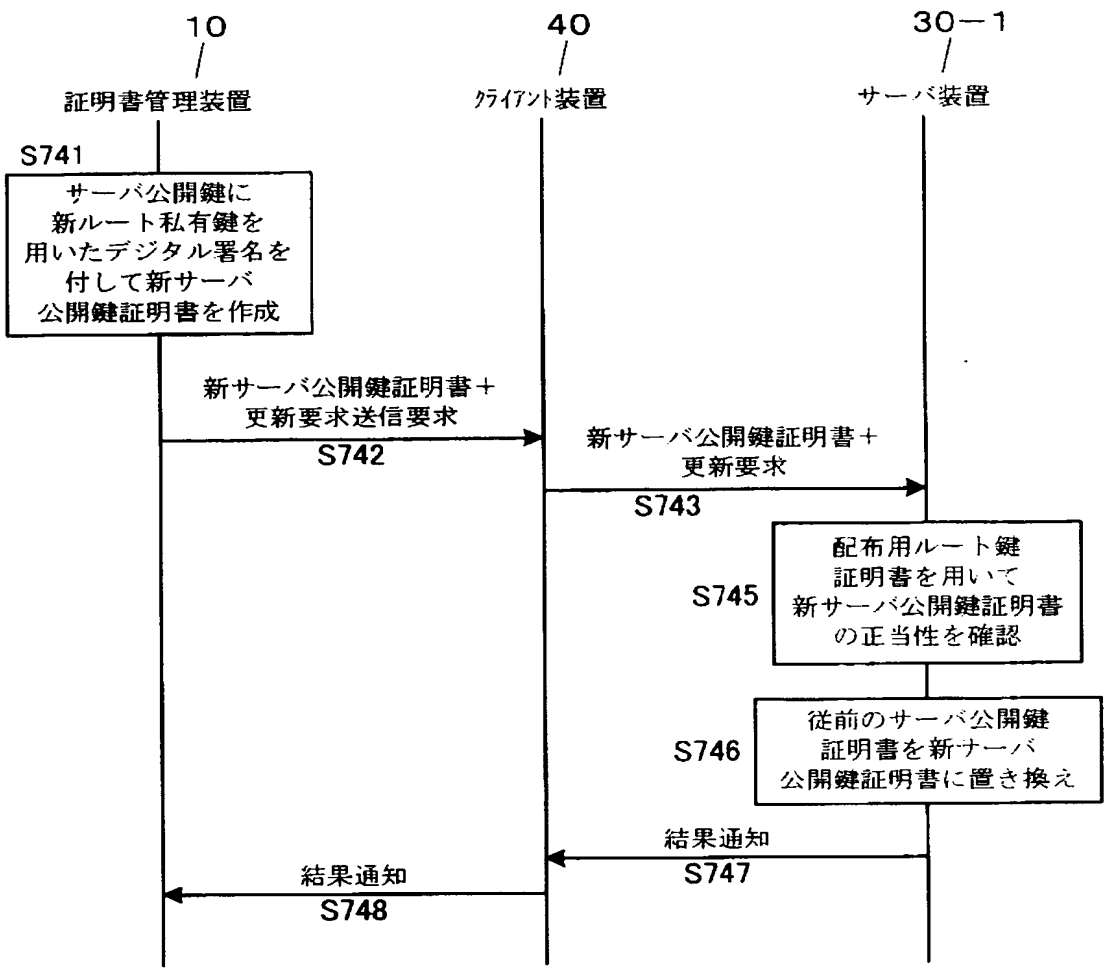
（ｂ）

サーバ装置３０－１	
直接通信否	
なし	

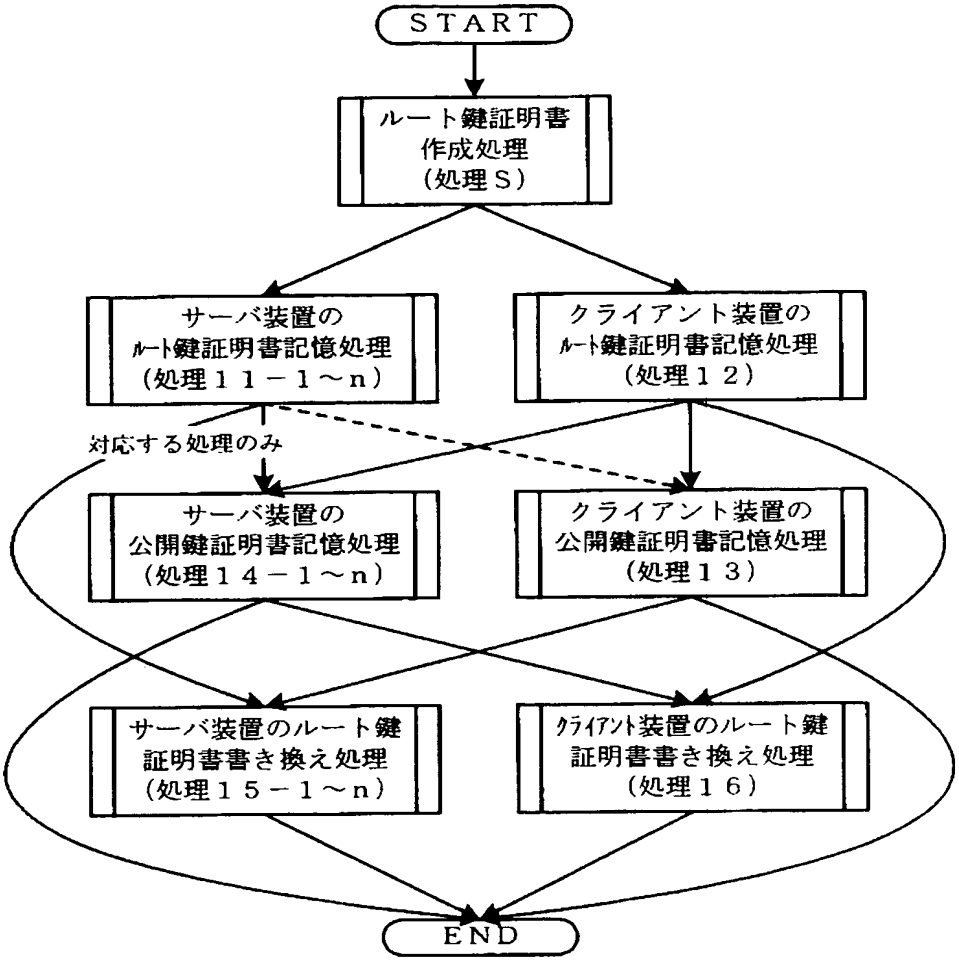
（ｃ）

サーバ装置３０－１	
直接通信否	
ｸﾗｲｱﾝﾄ装置３０	サーバ
	ルート鍵Ａ
	更新要

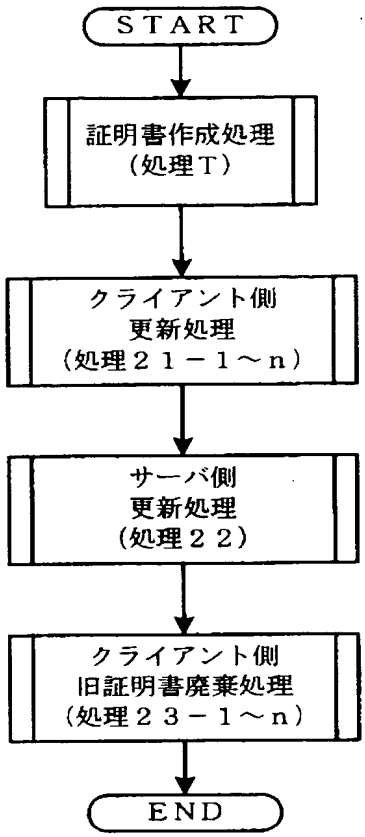
【図３７】


処理14－1
10
証明書管理装置
40
ｸﾗｲｱﾝﾄ装置
30－1
サーバ装置
S741
サーバ公開鍵に
新ルート私有鍵を
用いたデジタル署名を
付して新サーバ
公開鍵証明書を作成
新サーバ公開鍵証明書＋
更新要求送信要求
S742
新サーバ公開鍵証明書＋
更新要求
S743
S745
配布用ルート鍵
証明書を用いて
新サーバ公開鍵証明書
の正当性を確認
S746
従前のサーバ公開鍵
証明書を新サーバ
公開鍵証明書に置き換え
結果通知
S747
結果通知
S748

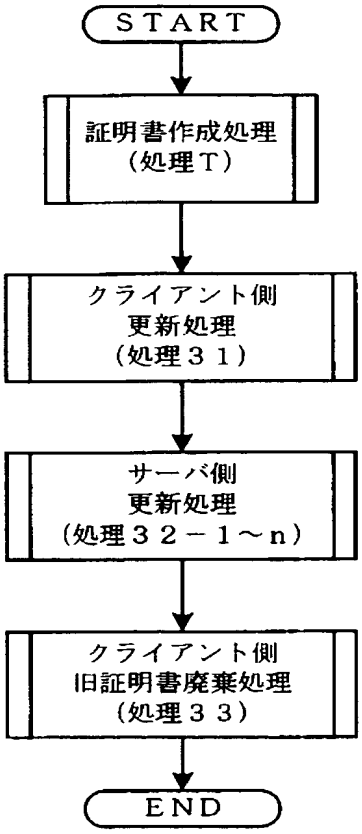
【図３８】


START
ルート鍵証明書
作成処理
（処理Ｓ）
サーバ装置の
ﾙｰﾄ鍵証明書記憶処理
（処理１１－１～ｎ）
クライアント装置の
ﾙｰﾄ鍵証明書記憶処理
（処理１２）
対応する処理のみ
サーバ装置の
公開鍵証明書記憶処理
（処理１４－１～ｎ）
クライアント装置の
公開鍵証明書記憶処理
（処理１３）
サーバ装置のルート鍵
証明書書き換え処理
（処理１５－１～ｎ）
ｸﾗｲｱﾝﾄ装置のルート鍵
証明書書き換え処理
（処理１６）
END

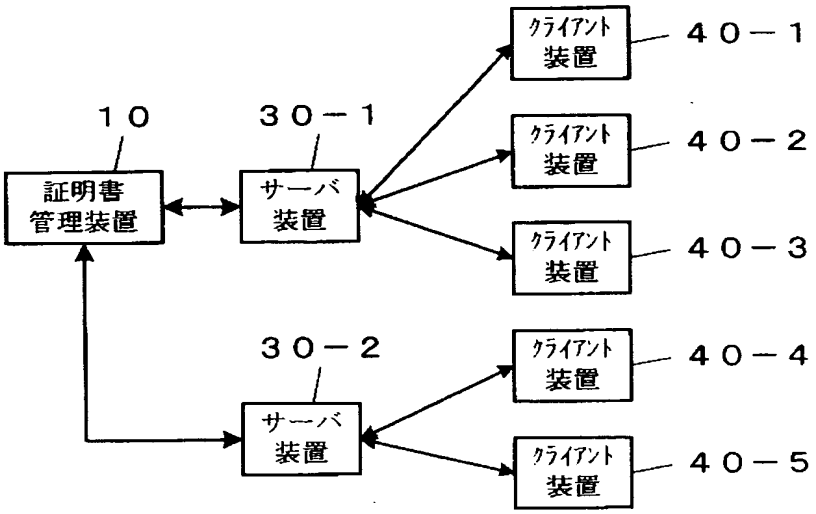
【図３９】


START
証明書作成処理
（処理T）
クライアント側
更新処理
（処理２１－１～ｎ）
サーバ側
更新処理
（処理２２）
クライアント側
旧証明書廃棄処理
（処理２３－１～ｎ）
END

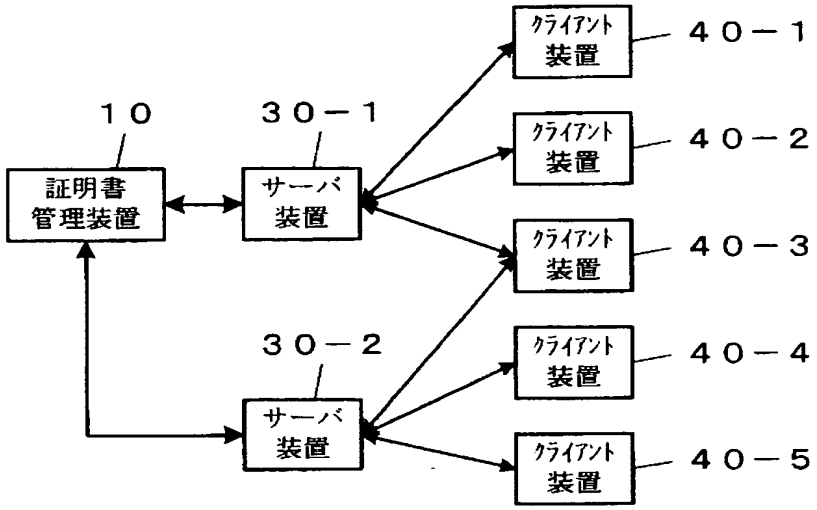
【図４０】


START
証明書作成処理
（処理T）
クライアント側
更新処理
（処理３１）
サーバ側
更新処理
（処理３２－１～n）
クライアント側
旧証明書廃棄処理
（処理３３）
END

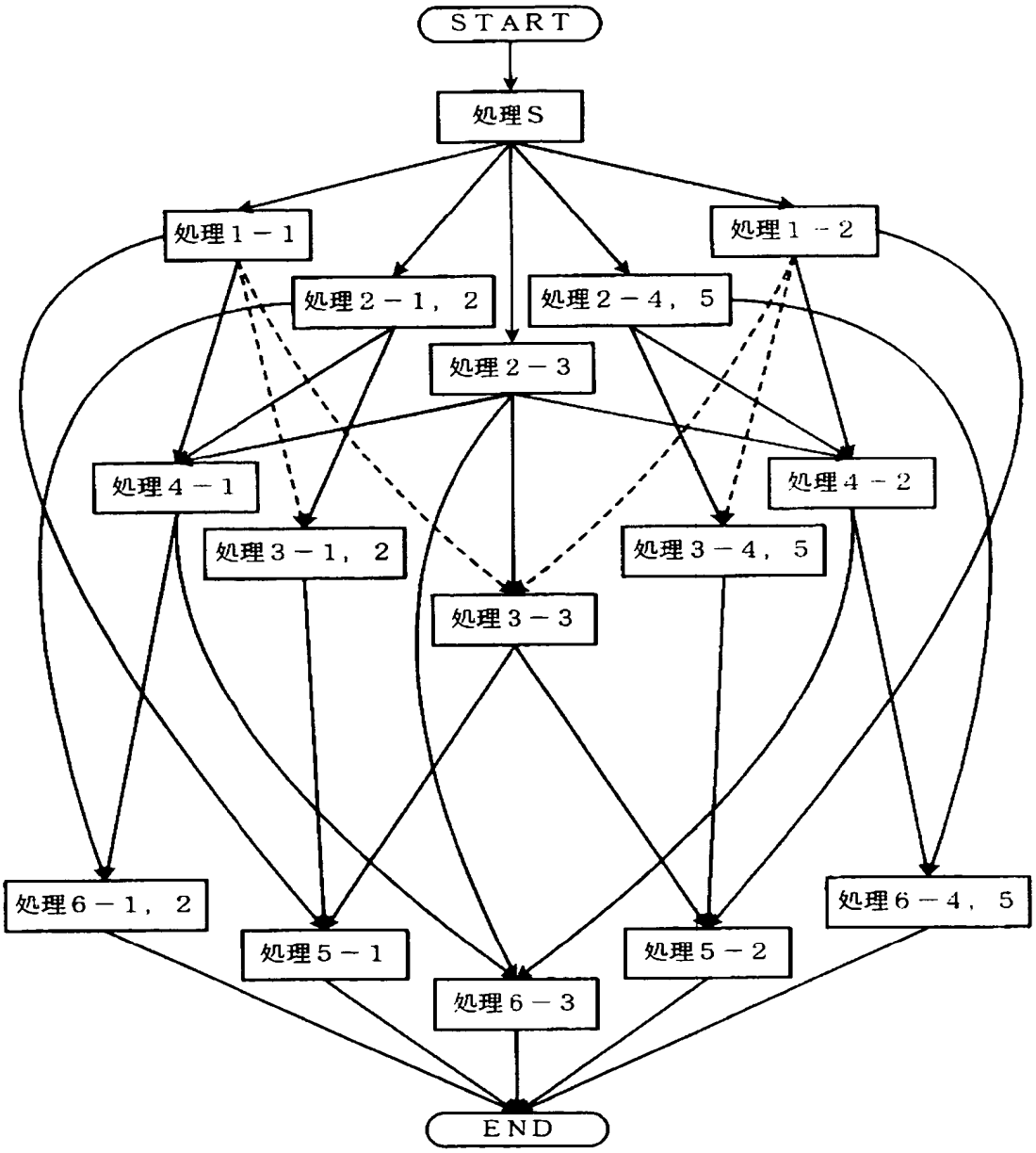
【図４１】


１０
証明書
管理装置
３０－１
サーバ
装置
３０－２
サーバ
装置
クライアント
装置
４０－１
クライアント
装置
４０－２
クライアント
装置
４０－３
クライアント
装置
４０－４
クライアント
装置
４０－５

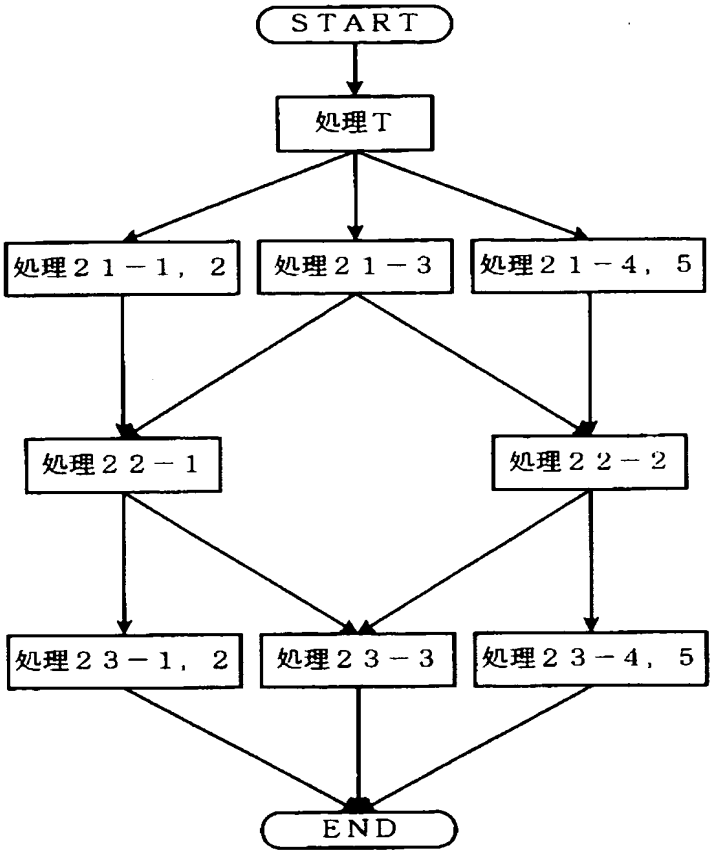
【図４２】


１０
証明書
管理装置
３０－１
サーバ
装置
３０－２
サーバ
装置
クライアント
装置
４０－１
クライアント
装置
４０－２
クライアント
装置
４０－３
クライアント
装置
４０－４
クライアント
装置
４０－５

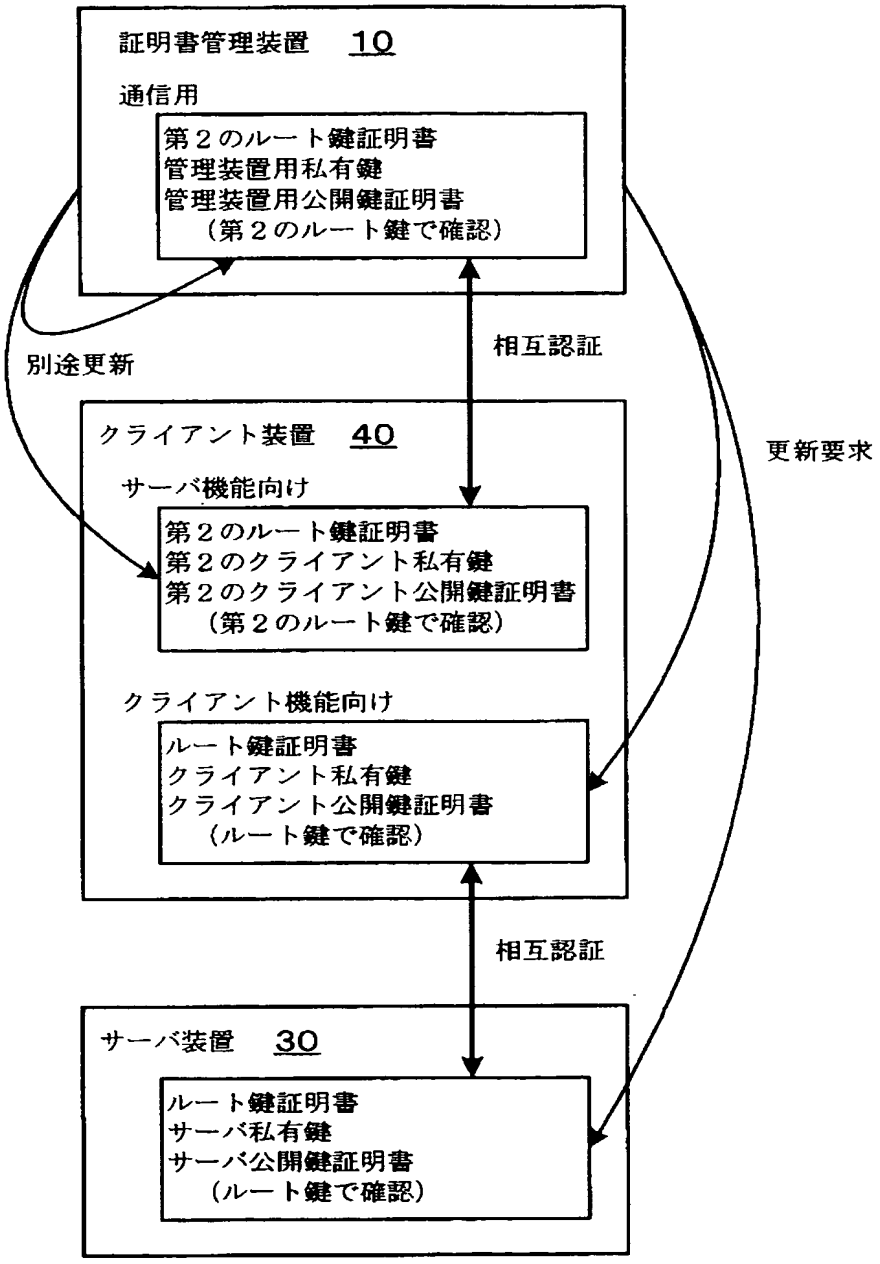
【図４３】


START
処理Ｓ
処理１－１
処理１－２
処理２－１，２
処理２－４，５
処理２－３
処理４－１
処理４－２
処理３－１，２
処理３－４，５
処理３－３
処理６－１，２
処理６－４，５
処理５－１
処理５－２
処理６－３
END

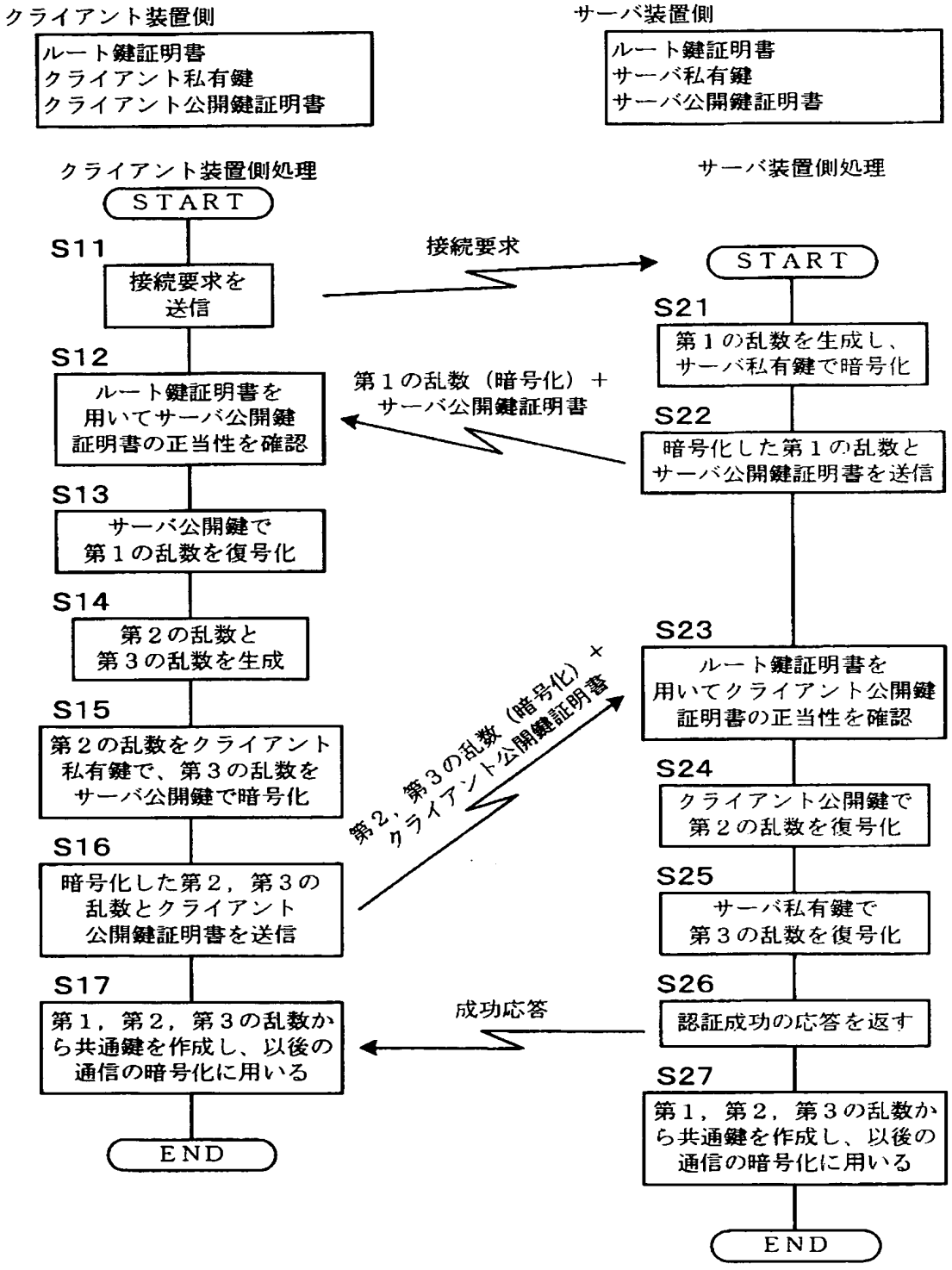
【図４４】


START
処理Ｔ
処理２１－１，２
処理２１－３
処理２１－４，５
処理２２－１
処理２２－２
処理２３－１，２
処理２３－３
処理２３－４，５
END

【図４５】

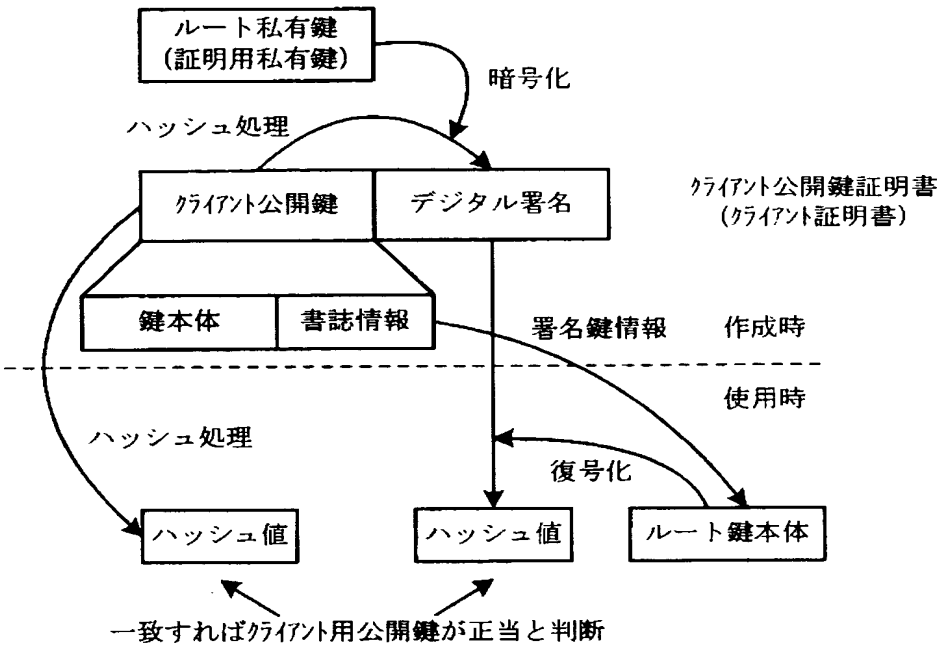

証明書管理装置　10
通信用
第２のルート鍵証明書
管理装置用私有鍵
管理装置用公開鍵証明書
（第２のルート鍵で確認）
相互認証
別途更新
クライアント装置　40
更新要求
サーバ機能向け
第２のルート鍵証明書
第２のクライアント私有鍵
第２のクライアント公開鍵証明書
（第２のルート鍵で確認）
クライアント機能向け
ルート鍵証明書
クライアント私有鍵
クライアント公開鍵証明書
（ルート鍵で確認）
相互認証
サーバ装置　30
ルート鍵証明書
サーバ私有鍵
サーバ公開鍵証明書
（ルート鍵で確認）

【図４６】

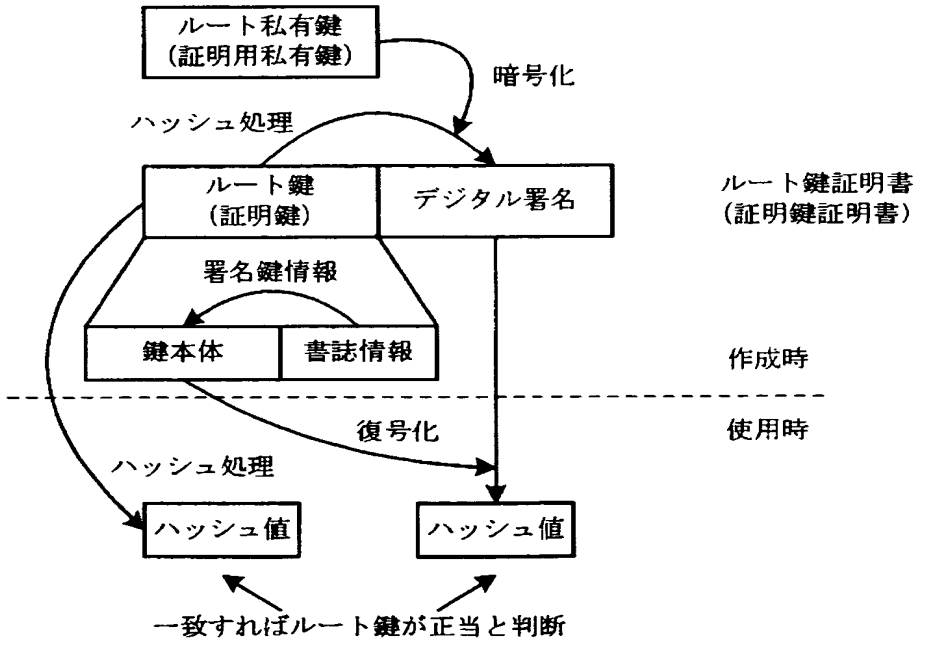

クライアント装置側
ルート鍵証明書
クライアント私有鍵
クライアント公開鍵証明書
サーバ装置側
ルート鍵証明書
サーバ私有鍵
サーバ公開鍵証明書
クライアント装置側処理
サーバ装置側処理
START
S11
接続要求を送信
接続要求
START
S21
第１の乱数を生成し、サーバ私有鍵で暗号化
S12
ルート鍵証明書を用いてサーバ公開鍵証明書の正当性を確認
第１の乱数（暗号化）＋サーバ公開鍵証明書
S22
暗号化した第１の乱数とサーバ公開鍵証明書を送信
S13
サーバ公開鍵で第１の乱数を復号化
S14
第２の乱数と第３の乱数を生成
S23
ルート鍵証明書を用いてクライアント公開鍵証明書の正当性を確認
S15
第２の乱数をクライアント私有鍵で、第３の乱数をサーバ公開鍵で暗号化
第２，第３の乱数（暗号化）＋クライアント公開鍵証明書
S24
クライアント公開鍵で第２の乱数を復号化
S16
暗号化した第２，第３の乱数とクライアント公開鍵証明書を送信
S25
サーバ私有鍵で第３の乱数を復号化
S17
第１，第２，第３の乱数から共通鍵を作成し、以後の通信の暗号化に用いる
成功応答
S26
認証成功の応答を返す
S27
第１，第２，第３の乱数から共通鍵を作成し、以後の通信の暗号化に用いる
END
END

【図４７】

（ａ）


ルート私有鍵
(証明用私有鍵)
暗号化
ハッシュ処理
ｸﾗｲｱﾝﾄ公開鍵
デジタル署名
ｸﾗｲｱﾝﾄ公開鍵証明書
(ｸﾗｲｱﾝﾄ証明書)
鍵本体
書誌情報
署名鍵情報
作成時
使用時
ハッシュ処理
復号化
ハッシュ値
ハッシュ値
ルート鍵本体
一致すればｸﾗｲｱﾝﾄ用公開鍵が正当と判断

（ｂ）


ルート私有鍵
(証明用私有鍵)
暗号化
ハッシュ処理
ルート鍵
(証明鍵)
デジタル署名
ルート鍵証明書
(証明鍵証明書)
署名鍵情報
鍵本体
書誌情報
作成時
使用時
復号化
ハッシュ処理
ハッシュ値
ハッシュ値
一致すればルート鍵が正当と判断

【書類名】　　要約書

【要約】

【課題】　クライアント・サーバシステムにおける認証処理でデジタル証明書の正当性の確認に用いるルート鍵を自動的に更新できるようにする。

【解決手段】　クライアント装置とサーバ装置との間で公開鍵暗号を利用したデジタル証明書を用いるＳＳＬ等の方式による相互認証を行うようにしたクライアント・サーバシステムに、デジタル証明書管理装置を接続し、サーバ装置とクライアント装置のルート鍵を自動的に更新するデジタル証明書管理システムを構成する。そして、この更新処理において、サーバ装置の公開鍵証明書を更新する処理（処理４）を、そのサーバ装置の通信相手となる全てのクライアント装置について新ルート鍵を記憶させる処理（処理２－１～ｎ）が完了した後で行うようにする。

【選択図】　　図３１

特願２００３－０９６１２９

出　願　人　履　歴　情　報

識別番号　［０００００６７４７］

１．変更年月日　２００２年　５月１７日
［変更理由］　住所変更
住　所　東京都大田区中馬込１丁目３番６号
氏　名　株式会社リコー